http://www.oldgamesitalia.net/

N. 32/1991
Sped. in abb. post. gr. III/70

L. 15.000

68000

AMIGA BYTE

by Elettronica 2000

SUL DISCO

CROSSWORD LE PAROLE INCROCIATE
PAC-COPY GIOCA DURANTE LA COPIA
POWERBOOT METTE UN MENU NEL BOOTBLOCK
TEXTCALC I CALCOLI SUI TESTI
SPACE VADER SPACE ARCADE ACTION
TEXTRA EDITOR PER SCRIVERE TESTI
BOOTLETTER UN BOOT COLORATISSIMO
BENCHBALL GIOCA A PALLA NEL WORKBENCH
JMENU PERSONALIZZA I TUOI SCRIPT
MEMLOOK UNO SGUARDO ALLA MEMORIA
MINIMAZE CREA E RISOLVI ROMPICAPO
BOUNCEVIEW DUE SLIDESHOW RIMBALZANTI

Grafica 3D
IMAGINE: RAY TRACING A 24 BIT

TIPS & TRICKS

DeskTop Video
SCALA: SLIDE SHOW E TITOLAZIONI ANIMATE

Programmazione
LE FUNZIONI

Linguaggi
BENCHMARK MODULA 2

Tools
RESOURCE 4.0 HYPERBOOK SISTHEMA PLUS

Megagame
TUTTI I CODICI DI LEMMINGS

Applicativi
GOLD DISK

http://www.oldgamesitalia.net/
http://www.oldgamesitalia.net/

http://www.oldgamesitalia.net/

AMIGA BYTE

Direttore
SIRA ROCCHI

Direzione Editoriale
MARIO MAGRONE

Direzione Tecnica
GIANCARLO CAIRELLA

Segreteria di Redazione
SILVIA MAIER

Grafica
NADIA MARINI

Fotografie
MARIUS LOOK

Disco a cura di
VITTORIO FERRAGUTI

Copertina
GIGI MURATORE

Collaborano ad AmigaByte: Francesco Annoni, Luca Arienti, Laura Baricevic, Paolo Bozzo, Luca Brigatti, Marco Brovelli, Paolo Colombo, Antonio De Lorenzo, Enrico Donna, Enrico Frascati, Renato Grossi, Fabrizio Lodi, Silvia Malaguti, Vincenzo Marangoni, Dario Martinelli, Luca Mirabelli, Pierluigi Montanari, Lorenzo Orlandini, Domenico Pavone, Graziano Pavone, Roberto Pellagatti, Riccardo Premoli, Guido Quaroni, Fabio Rossetti, Giuseppe Sacchi, Emanuele Scribanti, Paolo Sisti, Leonardo Tennozio, Aurora Tragara, Vertigo.

Redazione
C.so Vitt. Emanuele 15
20122 Milano
tel. 02/795047
ore pomeridiane.
Per telefonate tecniche: solo
mercoledì h 15-18.

Amministrazione, Redazione, Pubblicità: L'Agorà srl: C.so Vittorio Emanuele 15, 20122 Milano. Fotocomposizione: Compostudio Est, selezioni colore e fotolito: Eurofotolit. Stampa: Garzanti Editore S.p.A. Cernusco S/N (MI). Distribuzione: SO.DI.P. Angelo Patuzzi spa, Via Zuretti 25, Milano. Amiga Byte è un periodico mensile registrato presso il Tribunale di Milano al n. 215 il 29 marzo 1988. Resp. Sira Rocchi. Spedizione in abbonamento postale Gr. III/70. Pubblicità inferiore al 70%. Tutti i diritti sono riservati per tutti i paesi. Manoscritti, disegni, fotografie e programmi inviati non si restituiscono anche se non pubblicati. © 1991. Amiga è un marchio registrato Commodore. AmigaByte è una pubblicazione indipendente, non connessa in alcun modo con la Commodore Business Machines USA.

4 - GOLD DISK OFFICE
10 - MODULA 2
15 - IL MEGLIO DEL PD
17 - LETTERE
23 - I CODICI DI LEMMINGS
26 - IMAGINE
33 - TIPS & TRICKS
34 - TOOLS
40 - LE FUNZIONI DI ARP
45 - SOFTWARE EXPRESS

IL MENU

http://www.oldgamesitalia.net/
http://www.oldgamesitalia.net/

http://www.oldgamesitalia.net/

SUL DISCHETTO...

La linea di confine che separa le utility dai giochi nel dischetto di questo mese non è semplice da definire come al solito: alcuni dei programmi che

AmigaByte propone questa volta possono infatti essere facilmente catalogati in entrambe le categorie.

PAC-COPY è un tipico esempio: si tratta di un copiatore di dischetti molto potente e veloce, derivante come concezione dal diffuso RattleCopy. La peculiarità consiste nel fatto che, durante la duplicazione dei dischi, non vi annoierete più osservando il numero di settori che viene copiato poiché nella metà inferiore dello schermo potrete giocare ad una versione personalizzata del celebre gioco Pac Man.

Anche **CROSSWORD** e **MINIMAZE** sono giochi con una marcia in più: il primo vi sfida a completare schemi di parole crociate, mentre il secondo è un classico labirinto tridimensionale di dimensioni ridotte, adatto per essere giocato in multitasking durante le pause di esecuzione di qualche altro programma. Entrambi i giochi sono dotati di un editor interno che vi consente di personalizzarli e di creare a piacimento nuovi enigmi e nuovi labirinti da completare o sottoporre agli amici.

L'angolo riservato ai passatempi si esaurisce con **BENCHBALL,** una spassosa versione del gioco Pong ambientata nello schermo del WorkBench, e con **SPACE VADER,** un classico gioco arcade nella tradizione degli shoot'em-up di ambientazione spaziale.

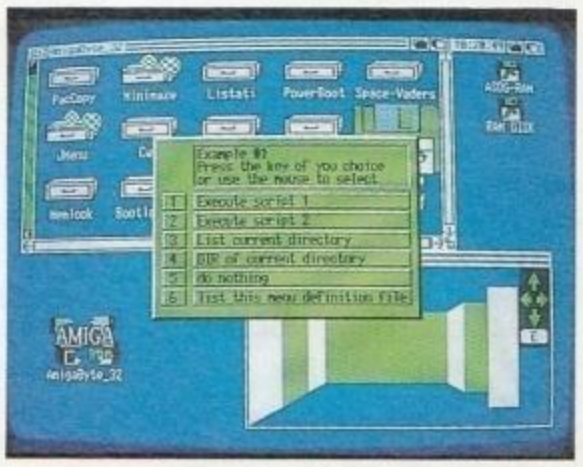
Se i giochi di questo mese sembrano utility, anche alcune utility sono decisamente meno serie del solito: è il caso di **BOUNCEPIC** e **BOUNCESHOW,** due tool che visualizzano immagini Iff facendole rimbalzare vistosamente sullo schermo. Il primo programma carica immagini singole, mentre il secondo può essere usato per la realizzazione di veri e propri slideshow e può riprodurre anche suoni campionati come sottofondo alle immagini.

L'utilizzo più immediato di **JMENU** è nella startup-sequence dei dischetti; grazie a questo tool potrete realizzare degli script interattivi che eseguiranno programmi in base alle scelte effettuate dall'utente tramite il mouse.

Quante volte vi è capitato di dover scrivere una lettera o una fattura, e di dover effettuare dei conti a mente o con una calcolatrice?

TEXTCALC risolve il problema, permettendo di effettuare i calcoli prelevando direttamente dal testo i numeri da elaborare, senza bisogno di doverli digitare manualmente. Non sapete nemmeno come fare per scrivere l'ipotetica fattura? Potete usare **TEXTRA,** un semplice e pratico editor per la stesura e la stampa di testi.

Gli smanettoni non sono stati dimenticati: a loro sono dedicate le tre restanti utility del dischetto.



MEMLOOK visualizza graficamente il contenuto della memoria di Amiga, permettendo di individuare immagini Iff, forme d'onda di suoni campionati, etc.

BOOTLETTER inserisce nel bootblock scritte scorrevoli con sfondo multicolore: il bootblock del dischetto di AmigaByte viene realizzato proprio grazie a questa utility, e rappresenta un esempio delle sue capacità.

Infine **POWERBOOT** consente di realizzare dischetti-compilation di programmi, inserendo nel bootblock un menu che funga da caricatore: all'utente sarà richiesto solo di effettuare una semplice selezione con il mouse per eseguire il programma prescelto.

http://www.oldgamesitalia.net/
http://www.oldgamesitalia.net/

http://www.oldgamesitalia.net/

Gold Disk Office

Un pacchetto integrato contenente cinque nuove versioni di celebri programmi della Gold Disk per scrivere, archiviare, calcolare, produrre grafici e persino dedicarsi al DeskTop Publishing.

di LUCA BRIGATTI

Una delle critiche ad Amiga più difficile da controbattere è quella secondo la quale questo computer, pur essendo valido sotto l'aspetto grafica - animazione - videogiochi, lasci piuttosto a desiderare quando si parla di applicazioni professionali di altro tipo. In altre parole, pochi hanno dubbi sul computer da scegliere se si pensa di fare prevalentemente grafica o se si pensa ad un computer come *game machine*, ma quanti acquisterebbero un Amiga da tenere in ufficio? Quanti, pensando a videoscrittura, archiviazione o contabilità sceglierebbero Amiga?

Per colmare questa diffidenza nei confronti di un computer che pure dispone di molti pacchetti software del tipo citato, (molti dei quali recensiti e descritti in queste stesse colonne nei fascicoli precedenti) la Gold Disk ha introdotto sul mercato «**Office**», un pacchetto integrato composto di quattro dischi. Per pacchetto integrato s'intende una serie di programmi di vario tipo, più o meno collegati fra di loro, e «Office» non fa eccezione, essendo composto da cinque programmi diversi: «**Write**», un word processor; «**Calc**», un foglio elettronico; «**Graph**», un generatore di grafici, strettamente collegato a «Calc»; «**File**», un database, e «**Page**», un programma di DeskTop Publishing. Completano il tutto due utility: «**CacheEditII**», che lavora in congiunzione con «Page», e «**Spell**», uno *spelling checker* da utilizzarsi insieme a «Write».

PRO E CONTRO DEI PACCHETTI INTEGRATI

L'idea non nuova dei pacchetti integrati presenta vantaggi e svantaggi, ai quali «Office» non si sottrae: il vantaggio sicuramente più evidente è la possibilità di risolvere i problemi di un ufficio con l'acquisto di un solo pacchetto software, con la non secondaria prerogativa di contenere i costi. Un altro vantaggio consiste nella maggior facilità di scambiare file fra un pacchetto e l'altro che, nel caso di «Office», si traduce nella possibilità per «Page» di caricare ed utilizzare testi scritti con «Write», o addirittura di chiamare ed utilizzare «Write» come editor dall'interno di «Page». Oppure, per fare un altro esempio, i database di «File» possono essere tranquillamente letti da «Calc» e da «Graph». Il rovescio della medaglia consiste nel fatto che, pur trattandosi di programmi tutti validi, essi non

http://www.oldgamesitalia.net/
http://www.oldgamesitalia.net/

http://www.oldgamesitalia.net/

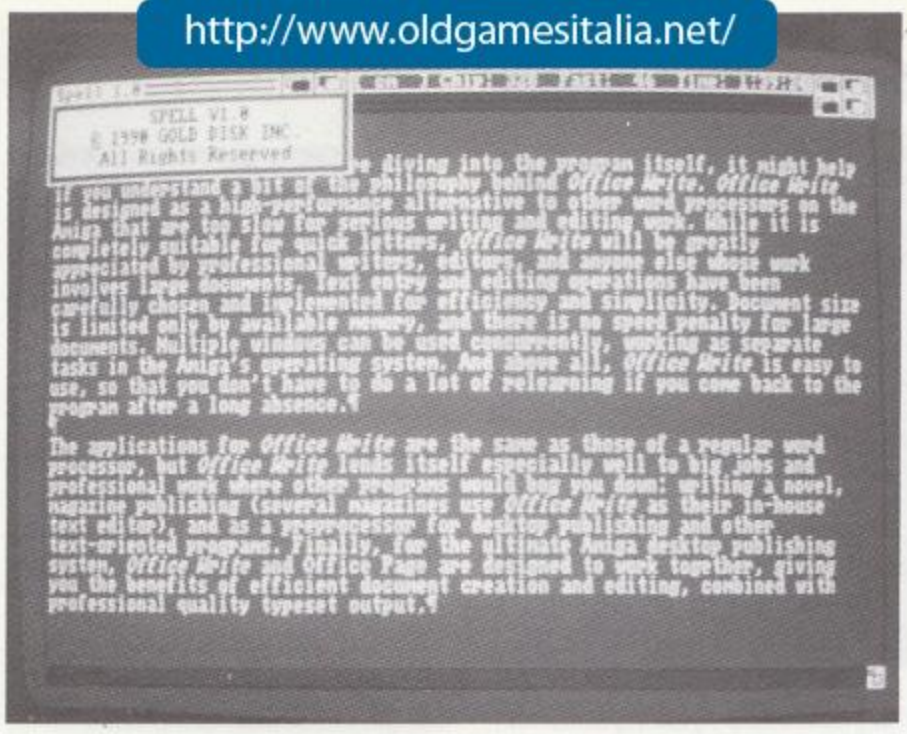
sono per forza di cose i migliori della loro categoria. O meglio, non tutti necessariamente rispondono esattamente alle esigenze di un determinato utente.

Il problema non si porrebbe acquistando i programmi separatamente, nel qual caso si potrebbe scegliere il miglior word processor (o il più adatto), il miglior foglio elettronico, il miglior database e così di seguito, con un concomitante lievitare di prezzi ed il rischio della non compatibilità dei file generati dai vari programmi.

Detto per inciso, i programmi proposti dalla Gold Disk in questo pacchetto integrato sono, per così dire, «reincarnazioni» di programmi già proposti separatamente dalla stessa software house: «Write» (il word processor) è, per esempio, una riedizione di «Transcript», mentre «Calc», il foglio elettronico, è una riedizione di «Advantage», e «Page» è un parente stretto di «PageSetter II».

Occorre precisare tuttavia, che non si tratta di un mero cambiamento di nomi a programmi preesistenti, ma che quelli contenuti in «Office» costituiscono delle edizioni aggiornate, ed in alcuni casi semplificate, dei programmi precedenti.

Amiga Byte ha recensito sul fascicolo scorso «**Advantage**», dunque in questa sede ci limiteremo a puntualizzare le differenze fra «Calc» e questo programma, rimandando chi fosse interessato all'articolo relativo ad «Advantage». Ma vediamo gli altri programmi.



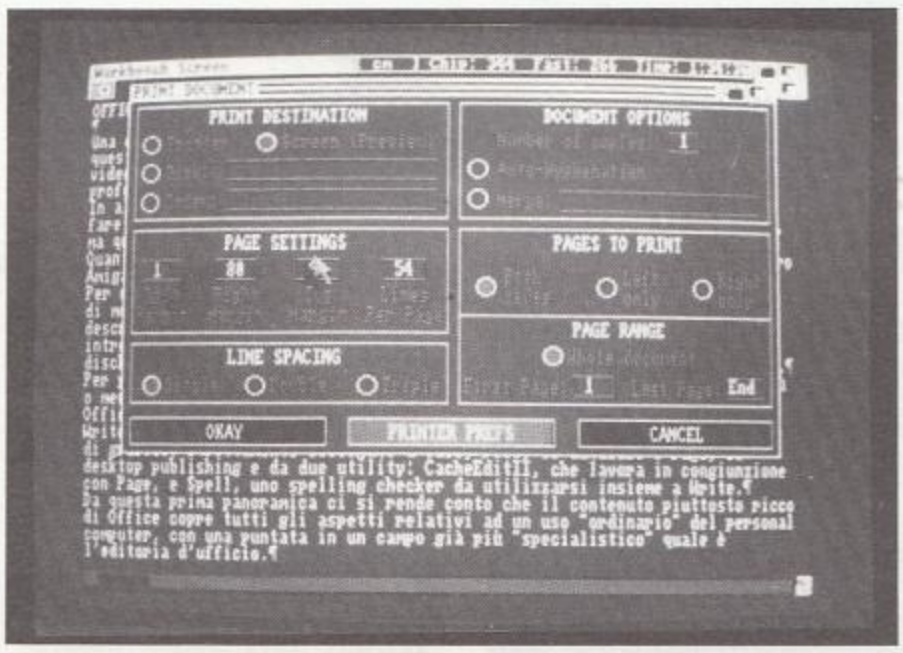
Lanciando l'utility Spell, il dizionario viene copiato in RAM per aumentare la velocità dello spelling checker.

IL PROGRAMMA WRITE

Caricando «**Write**», esso si apre in una finestra nello schermo del WorkBench, cosa comune anche a quasi tutti gli altri programmi, il che contribuisce a risparmiare memoria. I colori dell'ambiente di «Write» sono dunque identici a quelli del WorkBench, mentre la loro assegnazione (colore dello sfondo, dei testi, etc...) può essere modificata tramite l'opzione **Set Options** del **menu Special**, corrispondente anche alla combinazione di tasti Shift-F7. A questo proposito, una delle molte caratteristiche interessanti di questo software è la possibilità di richiamare tutte le funzioni dei menu con una combinazione di tasti; ciò rende molto più veloce l'uso del programma, una volta apprese le varie combinazioni.

Altra peculiarità di «Write» è quella che consente di aprire più finestre contemporaneamente (**Project / New Window** o Amiga Destro - T); il testo presente in una finestra può essere liberamente trasferito o copiato nelle altre finestre, rendendo così molto semplice eseguire sofisticate operazioni di editing.

Sempre nel menu Project, oltre che le tradizionali operazioni **Load** e **Open** (che differiscono fra loro in quanto Load carica un testo nella finestra corrente, Open in una nuova), **Save** e **Save as** (che differiscono per il fatto che la prima salva il testo con il nome e sul drive precedentemente specificato, la seconda richiede di fornire questi dati ogni volta), abbiamo le opzioni **Insert**, che aggiunge un testo a quello corrente a partire dalla posizione del cursore, e **Save Highlight Region**, che salva soltanto la zona di testo selezionata con il mouse.



Il menu Print di «Write» consente di impostare tutte le opzioni di stampa, fra cui anche la redirezione su video.

Load Macros e **Save Macros** consentono di caricare e salvare set di macro, ovvero serie di operazioni precedentemente assegnate ad una combinazione di tasti, caratteristica che spesso manca in WP anche più sofisticati.

L'OPZIONE PRINT

Un discorso a sé merita l'opzione **Print**, del **menu Project**. Come è intuibile, grazie a questo menu si possono pilotare tutte le operazioni relative alla stampa del testo in questione. Fra le opzioni, quella di indirizzare l'output su schermo ed avere così la possibilità di vedere in anteprima come esso apparirà una volta stampato.

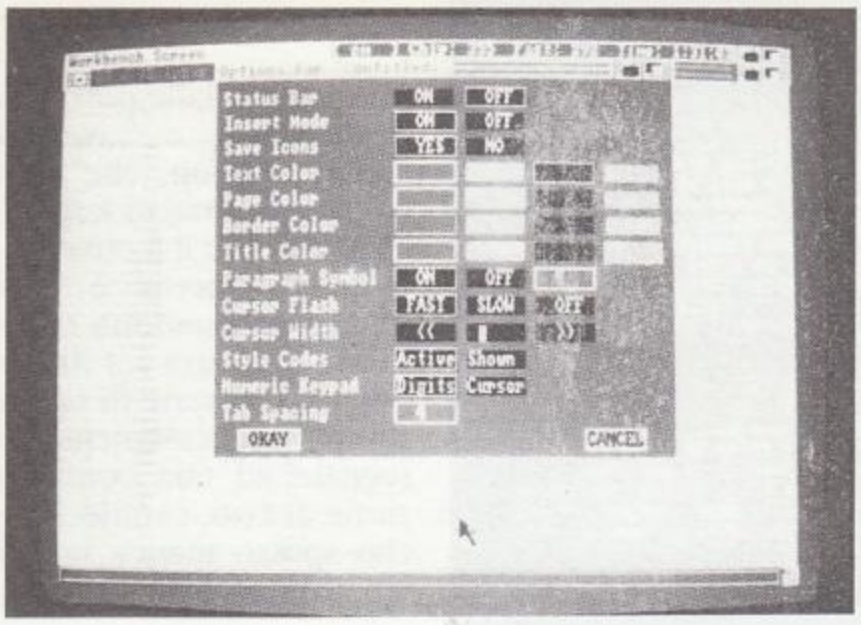
Il **menu Edit** pilota tutte le opzioni di «taglia e incolla» sulle quali non ci soffermeremo; basti dire che le opzioni sono numerose, e che velocizzano e semplificano di molto l'editing dei testi.

Il **menu Commands** pilota tutta una serie di operazioni fra cui quella di ricerca e di sostituzione, di trasformazione di un testo da maiuscolo a minuscolo e viceversa (**Convert to Lower/Upper Case**), di identificazione di parole che andranno a costituire un indice analitico (**Mark for Index**), nonché le operazioni relative al controllo ortografico (per la lingua inglese) del testo, per effettuare il quale occorre aver prima caricato l'utility «**Spell**», presente nello stesso disco; dopo di ché è possibile controllare una parola (**Spell-Check Word**), o tutto il testo dalla posizione del cursore in poi (**Spell-Check to End**).

Se una parola non corrisponde a nessuna di quelle

http://www.oldgamesitalia.net/
http://www.oldgamesitalia.net/

http://www.oldgamesitalia.net/



Tramite l'opzione Set Options del menu Special è possibile definire alcune caratteristiche dell'interfaccia utente, come ad esempio i colori.

presenti nel dizionario (che viene caricato in RAM, per ragioni di velocità), si può chiedere al programma di indovinare l'esatta compitazione (**Guess Spelling**) o di mostrare le parole del dizionario che più si avvicinano a quella dubbia (**Scan Dictionary**). Nell'uno o nell'altro caso, la parola può essere sostituita con una di quelle proposte semplicemente clickando su una di queste.

L'utility «Spell» può essere lanciata per proprio conto e consente di controllare testi in lingua inglese presenti su disco, anche se non prodotti da «Write».

Il menu Special offre svariate funzioni fra le quali la già citata Set Options o la simpatica **Tiny Window**, che rimpicciolisce e mette «a riposo» la finestra attiva.

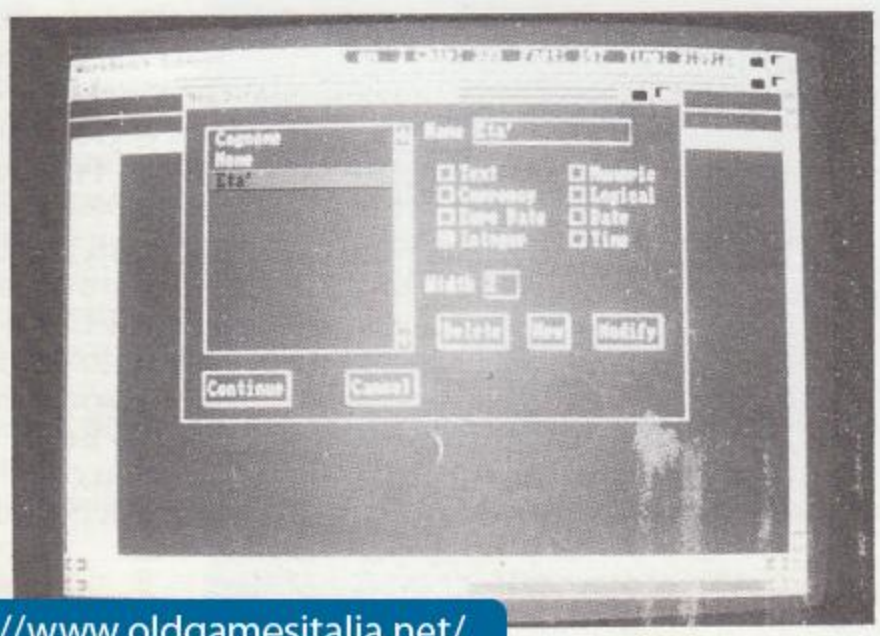
Il **menu Style** contiene tutte le opzioni necessarie per fare un po' di maquillage alle parole: grassetto, corsivo, sottolineato e così via.

Il **menu Cursor** offre innumerevoli opzioni (comode da utilizzare soprattutto tramite le sequenze di tasti) per spostare rapidamente il cursore nei punti più disparati del testo.

IL PROGRAMMA FILE

«**File**» è il database del pacchetto: ha la caratteristica di lavorare direttamente sul disco, senza tenere i dati in memoria: ciò significa che, a fronte di una minor velocità di esecuzione di alcune operazioni, non si corre il rischio di perdere dati per colpa di black out o di «guru» improvvisi.

**Ci sono cose che sono utili tutti i giorni.
Esempio di definizione dei campi di un ipotetico database di tipo «agenda telefonica».**



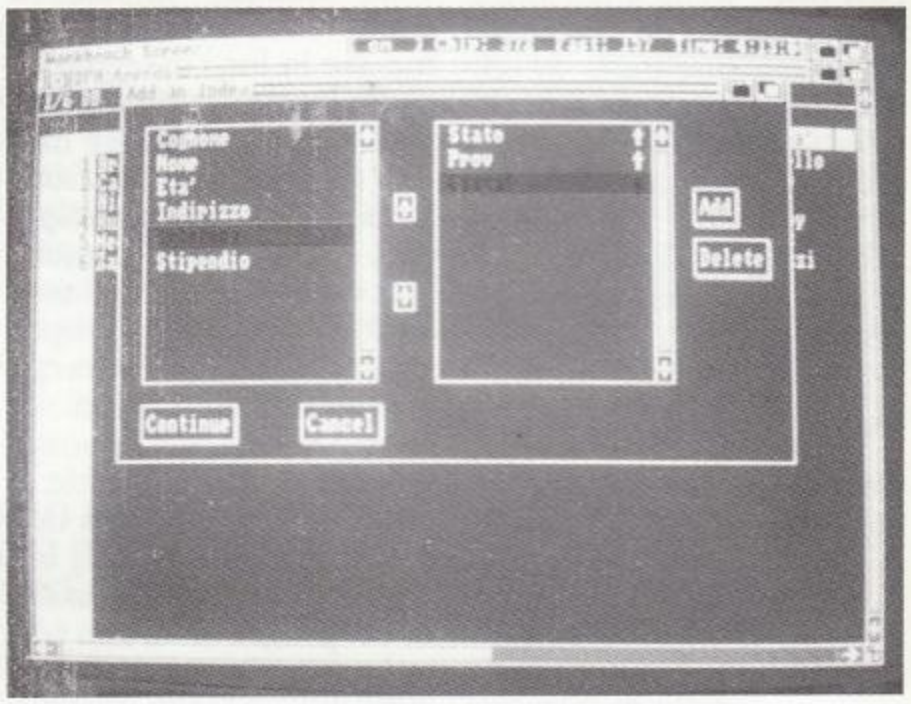
Al caricamento occorre specificare se si vuol creare un database nuovo o se si desidera caricarne uno già esistente. Per creare un database occorre assegnare un nome ai campi che si intendono utilizzare (ad esempio «Cognome», «Nome», «Indirizzo», etc...), nonché specificarne le caratteristiche. per caratteristiche si intendono il tipo di campo (Testo, Valuta, Data nel formato europeo, Intero, Numerico, Logico, Data e Ora) e la lunghezza (per i testi e per i numeri).

Fatto ciò, si procede a salvare la struttura del database tramite un requester visualizzato automaticamente dal programma. A questo punto si entra nel database vero e proprio. Ogni record viene visualizzato su di una riga, mentre i campi si trovano incolonnati (le colonne hanno la larghezza stabilita al momento della creazione del file).

La seconda fase dell'utilizzo di un database consiste nell'inserimento dei dati. Supponiamo di aver realizzato la struttura di un'agenda telefonica; inseriamo quindi i primi dati. L'opzione relativa è **Add** (Aggiungi) del **menu Record**.

Selezionandola appare un rettangolo rosso che delimita il primo campo. Digitiamo il primo dato (ad esempio il cognome) e premiamo Return. Il cursore passa al secondo campo, poi al terzo, e così di seguito finché non avremo riempito il record. Il cursore passa poi al primo campo di un nuovo record da inserire, e così via finché non si esce dalla fase inserimento deselezionando l'opzione Add.

I nuovi record vengono immediatamente ordinati in base al primo campo (in ordine alfabetico, numerico, per data, etc. a seconda del tipo di campo).



Per modificare l'ordinamento è necessario creare un indice specificando il campo su cui effettuarlo ed eventuali campi secondari.

Se si desidera avere un database ordinato in modo diverso, occorre creare un **Indice**, tramite l'opzione **Index/Create**. Dopo aver specificato il nome dell'indice (giacché per ogni database vi possono essere più indici, questi ultimi vanno identificati con un nome diverso) ci viene proposto un requester con i nomi dei campi del database.

Supponendo di volere un'agenda ordinata per

http://www.oldgamesitalia.net/
http://www.oldgamesitalia.net/

http://www.oldgamesitalia.net/

Stato di appartenenza dei soggetti che vi compaiono, clickeremo su «**Stato**» e poi sul **box Add**.

Se vogliamo dei sotto-ordinamenti (ad sempio, a parità di stato, un ordinamento per provincia e, a parità, di provincia, per città) potremo selezionare altri campi; nel nostro esempio, Prov e Città. Finita la selezione, si clicka sul **box «Continue»** per salvare il nuovo indice.

Per indicare al database di utilizzare un indice diverso da quello di default si seleziona **Index/Use** e, sul requester che appare, si clicka sull'indice desiderato.

condizione di filtratura. Per esempio, volendo vedere solo gli italiani, si inserirà la regola:

Stato="Italia"

Cercando il signor Rossi si inserirà:

Cognome="Rossi"

Volendo vedere solo i ricchi si selezionerà:

Stipendio>3000000

e così di seguito.

Selezionando **Filter/Apply** e clickando quindi sul filtro desiderato, si applica il medesimo al database, il quale funziona esattamente come prima tranne che non vengono visualizzati (vengono cioè filtrati) i record che non soddisfano le condizioni del filtro.

Creando ed applicando più filtri vengono visualizzati solo i record che soddisfano tutte le condizioni imposte dai vari filtri. Per togliere un filtro si usi: **Filter/Forget**.

Se non si desidera utilizzare i filtri, è possibile effettuare una ricerca con **Command/Find** che ci chiede di specificare una formula di ricerca (del tutto identica a quelle viste per i filtri) per poi andare a posizionare il cursore sul primo record che soddisfa quella condizione.

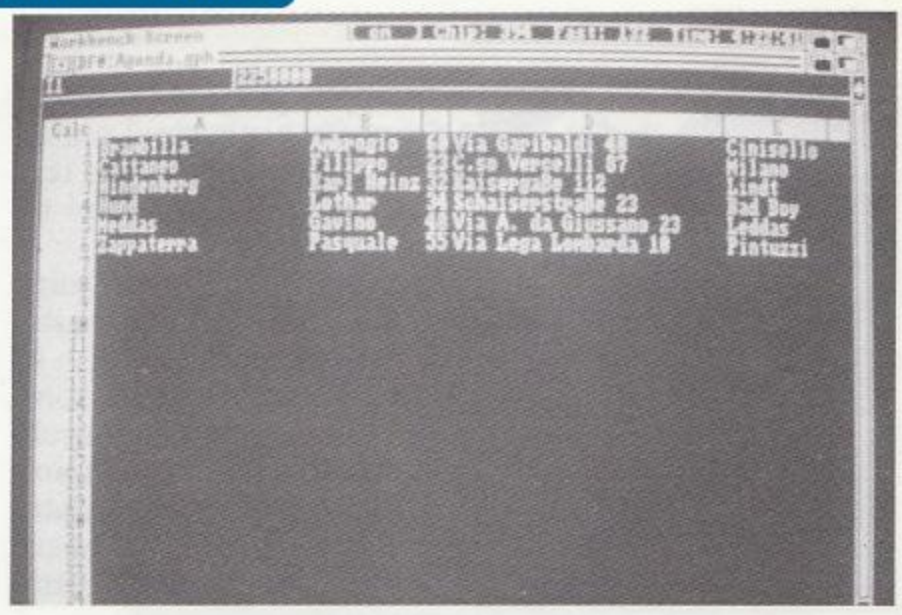
Per trovare il record successivo che abbia quelle caratteristiche si seleziona **Command/Find Next**.

Il modo di visualizzazione di default è **Browse**: se lo si deseleziona (**Project/Browse**) i record appaiono uno alla volta con i campi incolonnati.

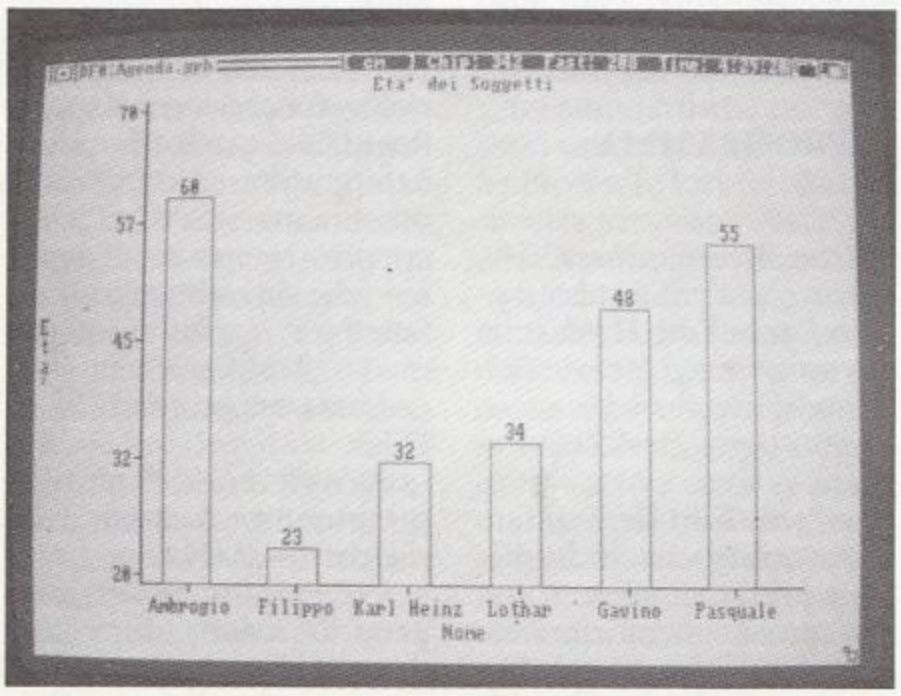
Si possono creare schermi personalizzati con **Screen/Create**, specificando le dimensioni di una pagina all'interno della quale è possibile scrivere qualsiasi cosa (ad esempio un'intestazione), modificando i nomi dei campi (ad esempio scrivere «Provincia» anziché «Prov», o «Nazione» anziché «Stato») e disponendo i campi ovunque nella pagina, posizionando il cursore e richiamando l'opzione **Screen/Add Field**, l'unica attiva in modo Screen/Create.

Per finire, «File» ci offre la possibilità di creare dei rapporti (**Report/Create**), ovvero disposizioni particolari dei vari campi su di una pagina, in funzione della stampa su carta. Specificato il nome del report, appare un requester che ci consente, utilizzando il mouse, di selezionare i campi da stampare, la loro posizione sulla carta, gli «a capo», i cambi pagina e le intestazioni dei vari campi. Dopo aver creato un report, i record del database possono essere stampati con il formato stabilito con **Report/Print**.

Sin qui abbiamo visto brevemente le caratteristiche essenziali di «File», ma ve ne sono altre piuttosto interessanti. Con **Project/Save as/Calc-Graph** è possibile salvare un database in modo che esso possa essere letto da «Calc», ad



Una delle peculiarità più interessanti di questo pacchetto integrato è la possibilità di caricare in «Calc» i database realizzati con «File».



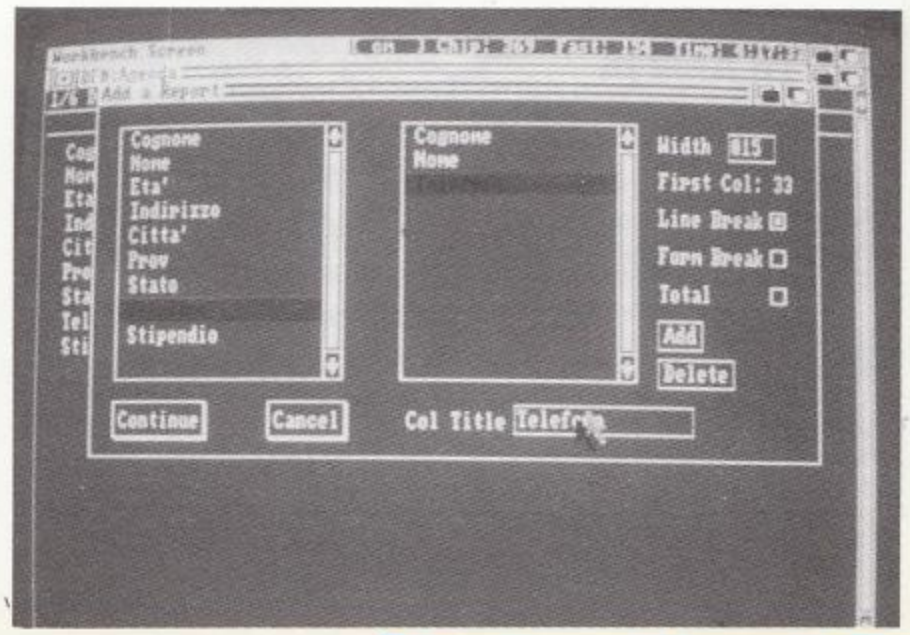
Esempio di grafico realizzato con «Graph» a partire dai dati caricati dal database «Agenda» di «File» e convertiti in formato «Calc».

LA RICERCA DEI RECORD

La terza fase dell'utilizzo di un database consiste nella ricerca dei record che ci interessano. Per eseguire qualsiasi ricerca, «File» mette a disposizione lo strumento potente dei **filtri**, grazie ai quali è possibile visualizzare solo quel o quei record che soddisfano un determinato criterio.

Un filtro si crea con il comando **Filter/Create**. Indicato il nome del filtro, si viene invitati ad inserire la

I record possono venir stampati in un «Report», per il quale occorre specificare i campi da stampare e la loro disposizione sul foglio di carta.



http://www.oldgamesitalia.net/
http://www.oldgamesitalia.net/

http://www.oldgamesitalia.net/

AMIGA
EXTASY

3 DISCHETTI !
LIRE 30.000

Una nuova
raccolta
di videogame
piccanti
e animazioni
... no comment!
per la tua
soft-teca
hardcore
strettamente
personale.

Un modo
diverso
di far fondere
il joystick
e di giocare
con il tuo
computer.

LE TENTAZIONI
DI AMIGA
Solo per adulti!

Per ricevere Amiga Extasy basta inviare vaglia postale ordinario di Lire 30.000 ad AmigaByte, C.so Vitt. Emanuele 15, Milano 20122. Specifica sul vaglia stesso la tua richiesta e il tuo indirizzo. Per un recapito più rapido aggiungi lire 3mila e chiedi spedizione espresso!

esempio per eseguire elaborazioni sui campi numerici, o da «Graph», per realizzare grafici a partire da questi ultimi.

Con **Command/Quick Calc** è possibile eseguire rapidamente calcoli su campi numerici quali somme, medie, conteggi, avere il valore massimo o minimo e addirittura la varianza e la deviazione standard dei numeri inclusi nel campo interessato.

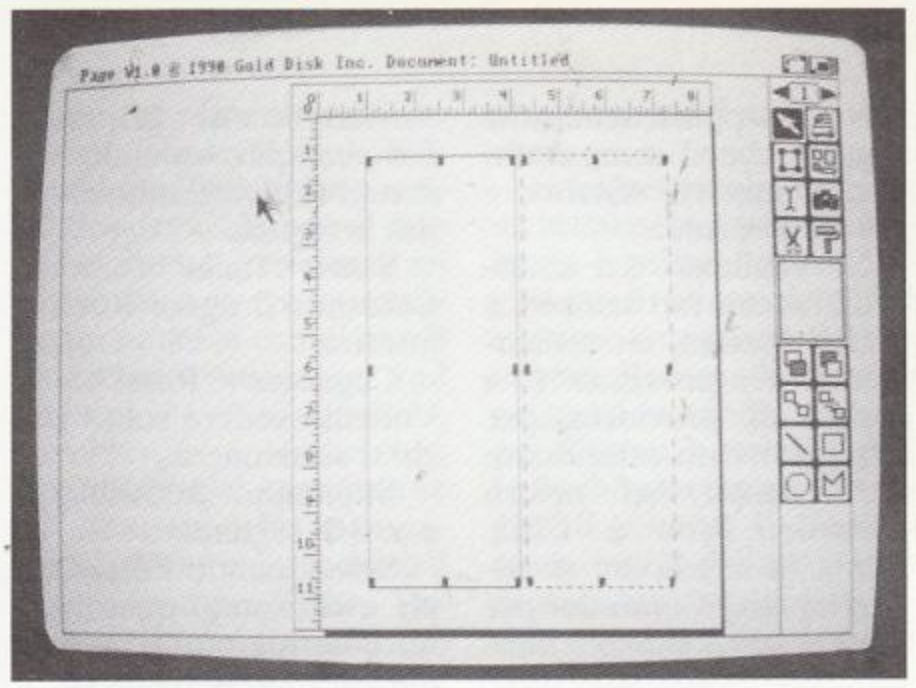
Citiamo infine la possibilità di interfacciare «File» con altri programmi tramite il linguaggio ARexx.

I PROGRAMMI CALC E GRAPH

«**Calc**» e «**Graph**» rappresentano la versione ridotta di «Advantage», che incorporava in un programma unico le capacità di calcolo e grafiche. Analogamente a «Write» ed a «File», e a differenza di «Advantage», i due programmi si aprono sulla finestra del WorkBench, sempre per ragioni di risparmio di memoria. Entrambi poi si aprono con un foglio elettronico vuoto che è possibile riempire, o nel quale è possibile caricare un foglio elettronico già realizzato.

«Calc» offre un menu più ricco ed è, fra i due, quello che viene utilizzato in pratica per realizzare e gestire un foglio elettronico ma, naturalmente, manca della possibilità di realizzare grafici. «Graph», al contrario, pur consentendo una certa possibilità di editing del foglio elettronico, ha un menu più ridotto di quello di «Calc» ma possiede il comando **New Chart** (**menu Command**) che introduce in un sottomenu tutto dedicato alla realizzazione di grafici.

Nelle loro funzioni essenziali, «Calc» e «Graph» non si discostano molto da «Advantage», al quale rimandiamo per una panoramica più completa. Citiamo soltanto la possibilità, anche per «Calc», di un interfacciamento con altri programmi tramite ARexx.



Per creare un documento con «Page», è necessario preparare una pagina, con il numero di «gabbie» desiderato.

IL PROGRAMMA PAGE

Con il programma «**Page**» e con l'utility accessoria «**CacheEdit II**» facciamo un salto nel mondo della «editoria da scrivania», meglio detta DeskTop Publishing.

«CacheEdit II» è soltanto un'utility che va impiegata prima di «Page» per caricare in memoria i caratteri desiderati nel formato CompuGraphic (quello di «Professional Page», per intenderci), in modo che essi risultino poi disponibili per «Page». Al caricamento viene richiesto di inserire in un drive un dischetto contenente almeno un font CompuGraphic.

L'ambiente di lavoro di «CacheEdit II» è costituito da uno schermo diviso in due parti: a sinistra abbiamo due requester, tramite i quali si selezionano un font ed un formato dei caratteri CompuGraphic; a destra si ha uno schermo sul quale compaiono i caratteri utilizzabili.

Dopo aver selezionato il font e le dimensioni desiderate sui requester a sinistra, si selezionano i caratteri che si intende impiegare sullo schermo a destra. Dato l'ok al programma, questi elabora e quindi carica in memoria i caratteri richiesti nel formato specificato, che possono quindi venire utilizzati da «Page». Al caricamento di «Page», appare un «piano di lavoro», che dovrebbe risultare familiare a chi ha avuto modo di utilizzare in precedenza un programma di DTP.

A differenza degli altri programmi presenti nel pacchetto di «Office», «Page» si apre in uno schermo proprio, a due colori; sulla destra vi è una colonna con una raccolta di icone rapprentanti altrettante funzioni; per creare un documento occorre aprire una pagina con le caratteristiche desiderate. Innanzitutto si selezioni l'opzione **Layout Tolls** del **menu Preferences**. Appare un requester che ci offre la possibilità di scegliere l'unità di misura fra pollici, centimetri e pica (un'unità di misura utilizzata in tipografia).

Dopo aver selezionato i centimetri, sia per la pagina che per il righello (**ruler**) si selezioni **Page/New** per aprire una nuova pagina. A questo punto compare un requester in cui vanno indicate le pagine da aprire, la continuità delle colonne (cioè se un testo che non

http://www.oldgamesitalia.net/
http://www.oldgamesitalia.net/

http://www.oldgamesitalia.net/

entra in una colonna deve continuare nella successiva), la dimensione della pagina, completamente definibile o selezionabile fra quattro possibilità standard (è consigliabile il formato A4, quello più utilizzato in Italia), i quattro margini, il numero di colonne, e lo spazio fra una colonna e l'altra.

LE COLONNE

Le colonne, o gabbie, si possono riposizionare semplicemente clickando su una di esse e trascinandola nella nuova posizione, mentre si possono ridimensionare clickando sui gadget presenti sui lati e sui vertici.

Il documento può essere visto in dimensioni reali o in scala. Per variare la scala si utilizza **Preference/Magnification**; con una scala di 33% o 50% il documento appare tutto in uno schermo ma risulta difficile e sovente impossibile leggere il testo; con una scala pari al 100% o al 200% il testo risulta perfettamente leggibile ma il documento è visibile soltanto in parte e, per esplorare una pagina, bisogna ricorrere al gadget rettangolare al centro del menu ad icone sulla destra dello schermo.

Per inserire un testo in una gabbia si seleziona l'icona rappresentante (nelle intenzioni dei programmatori) il cursore di un editor (il terzo dall'alto a sinistra), si clicka nella gabbia desiderata e si comincia a scrivere.

Più comodamente, se abbiamo un testo già preparato basta caricarlo con **Project/Import/Text** e quindi «incollarlo» (**Edit/Paste**) nella gabbia desiderata.

Il testo, oltre che essere in ASCII, può essere stato creato con i programmi «Write», «Word Perfect», «Scribble!», «Textcraft» e «Textcraft Plus».

Come per il testo, anche un'immagine può essere caricata nella gabbia desiderata, basta selezionare **Project/Import/Bitmap Graphic** e selezionare da disco, tramite il requester che appare, l'immagine desiderata, che viene inserita nella gabbia attiva.

La gabbia attiva si riconosce dalle altre perché il suo contorno è solido anziché tratteggiato.

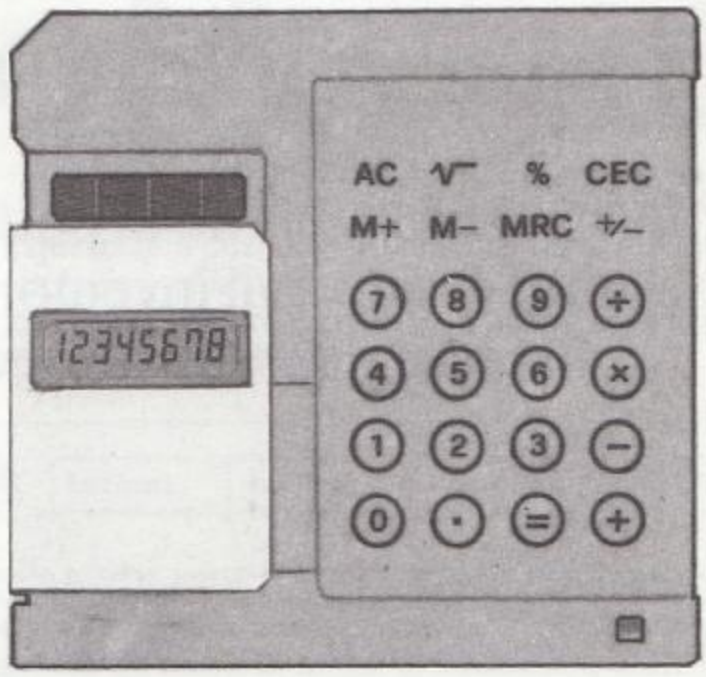
Queste poche informazioni dovrebbero già essere d'aiuto al lettore per acquistare famigliarità con questo programma, ma vale la pena di citare alcune altre funzioni offerte dal programma.

L'icona del menu grafico rappresentante una gabbia, la seconda dall'alto a sinistra, serve per creare nuove gabbie di dimensione a piacere. Il rullo s'impiega per creare automaticamente gabbie di dimensioni standard che poi possono essere ridimensionate a piacere; le forbici alla sua sinistra servono invece per eliminare la finestra attiva.

Le due icone rappresentanti riquadri sovrapposti, nella metà inferiore del menu ad icone, offrono la possibilità di spostare la gabbia attiva davanti o dietro le altre, quando due o più gabbie si sovrappongono parzialmente o completamente. Le due icone sottostanti hanno il compito rispettivamente di unire o separare più gabbie (in modo che il testo che non entra in una gabbia continui nella successiva). Per unire le gabbie si clicka sull'icona relativa, quindi sulle gabbie da unire, nell'esatta successione in cui si desidera che proceda il testo. I quattro gadget sottostanti consentono di tracciare semplici disegni.

«Page» offre la possibilità di stampare i documenti con una comune stampante a matrice di punti o su di una stampante laser con standard PostScript il che, unito alle caratteristiche dei font CompuGraphic, consente di ottenere da questo programma documenti di qualità eccellente.

NUOVISSIMA!
INSOLITA!
DIVERTENTE!
UTILE!



CALCOLATRICE-DISCO SOLARE

Ingegnosa, ha la forma e le dimensioni di un dischetto da 3.5 pollici.

☆

Così realistica che rischierete di confonderla nel mare dei vostri dischetti.

☆

Originale, praticissima, precisa, costa Lire 25.000, spese di spedizione comprese. In più, in regalo, un dischetto vero con tanti programmi... di calcolo.

☆

Per riceverla basta inviare vaglia postale ordinario di Lire 25 mila intestato ad AMIGA BYTE, c.so Vitt. Emanuele 15, 20122 MILANO. Indicate sul vaglia stesso, nello spazio delle comunicazioni del mittente, quello che desiderate, ed i vostri dati completi in stampatello. Per un recapito più rapido, aggiungete lire 3 mila e specificate che desiderate la spedizione Espresso.

http://www.oldgamesitalia.net/

http://www.oldgamesitalia.net/

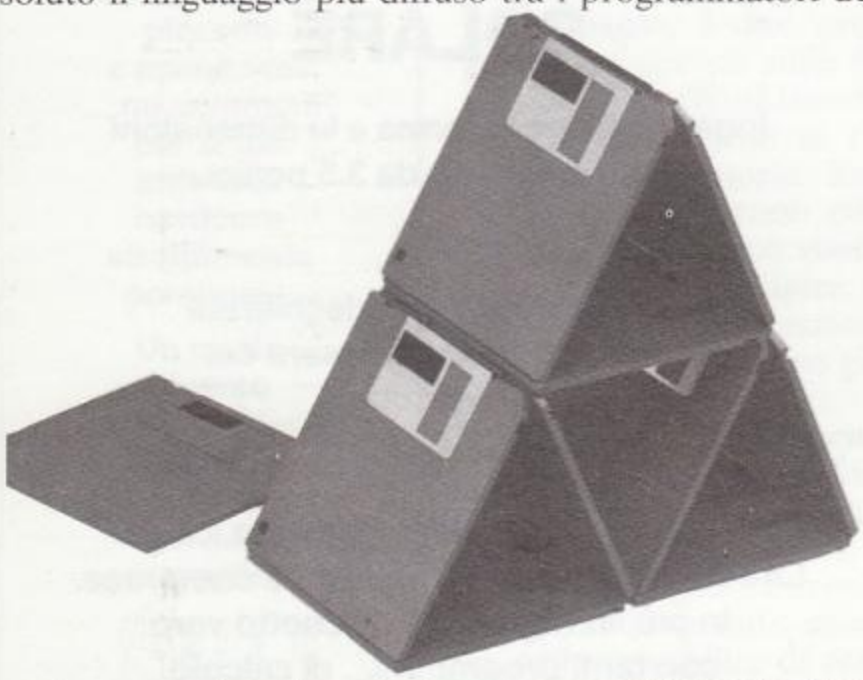
http://www.oldgamesitalia.net/

BenchMark Modula 2

Diamo uno sguardo al potente compilatore di un linguaggio per Amiga creato dall'inventore del Pascal, versatile quanto il C ma più facile da imparare, ed alle sue caratteristiche principali.

di PAOLO BOZZO
Prima parte

Per chi voglia programmare l'Amiga si prospettano varie possibilità di scelta nell'ambito dei linguaggi informatici e delle loro implementazioni: tra le diverse opportunità il **Modula 2**, per quanto non sia in assoluto il linguaggio più diffuso tra i programmatori del



nostro computer, si presenta come uno strumento dalle notevoli potenzialità e che deve senz'altro essere almeno preso in considerazione. Insomma, se intendete programmare l'*Amiga* seriamente e non avete ancora «sposato» un linguaggio particolare, sarà meglio valutare subito anche le caratteristiche del Modula 2 per non scoprire, dopo aver perso tempo a studiare il *C*, che quest'ultimo contrasta con la vostra *forma mentis* e che avreste preferito qualcosa di meno criptico e conciso, magari qualcosa di più simile al glorioso *Pascal*, che ha aperto la strada alla programmazione strutturata e che chi ha già lavorato su MS-DOS probabilmente conoscerà.

IL BENCHMARK MODULA 2

Il nostro intendimento non è comunque quello di sviluppare un corso sul *Modula 2*, ma di illustrare e commentare le caratteristiche del pacchetto «**Benchmark Modula 2**» pubblicato dalla Avant Garde e scritto da Leon Fraenkel. Si tratta di un compilatore non proprio recentissimo (1988), ma sviluppato con notevole professionalità e, in genere, piuttosto piacevole da usare. Pur non essendo privo di difetti, dispone di un *editor* molto completo, di un compilatore e di un *linker* integrati e piuttosto veloci, nonché di un ottimo *debugger* di sorgente. Offre insomma un ambiente completo (senz'altro migliore di quello del noto compilatore Modula 2 della TDI), un ambiente che in genere mette il programmatore a proprio agio, facilitandogli il lavoro di creazione, di sviluppo e di correzione dei programmi.

Nel contempo però, poiché del Modula 2 non si è mai parlato su questa nostra rivista, riteniamo opportuno offrire, in questa prima parte, alcuni chiarimenti molto generali sulle caratteristiche di questo interessante e potente linguaggio, in modo da dare al lettore che intenda ottenere il meglio dalla programmazione di *Amiga*, un utile aiuto nella scelta dello strumento che meglio gli permetta di realizzare le sue idee, i suoi progetti, di concretizzare la sua inventiva e fantasia in un prodotto soddisfacente.

UN LINGUAGGIO AD ALTO LIVELLO

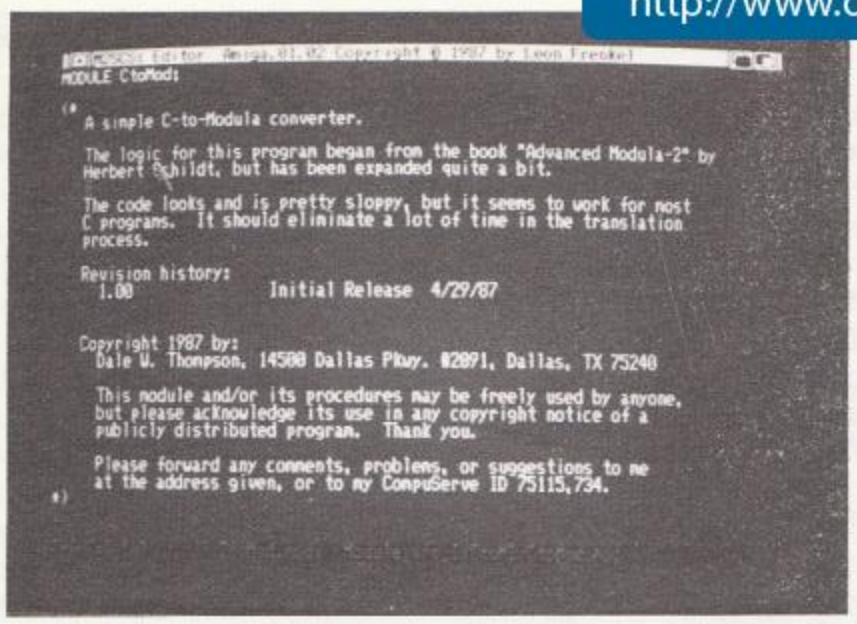
Come già osservato in passato, la scelta di un linguaggio è una faccenda molto personale e delicata. Gli algoritmi, le strutture, le *routine* che compongono un programma non sono altro che la formalizzazione del ragionamento umano e, se è giusto che si seguano le vie più efficienti ed economiche, è altrettanto giusto che ognuno di noi trovi la strada che ritiene più congeniale per risolvere i propri problemi di programmazione.

Una prima considerazione può essere fatta sul rapporto facilità/potenza di un linguaggio: in genere i due termini sono inversamente proporzionali, ma non è una legge così drastica da non ammettere eccezioni. Se utilizziamo

http://www.oldgamesitalia.net/

http://www.oldgamesitalia.net/

http://www.oldgamesitalia.net/


MODULE CtoMod;
(*
A simple C-to-Modula converter.
The logic for this program began from the book "Advanced Modula-2" by Herbert Schildt, but has been expanded quite a bit.
The code looks and is pretty sloppy, but it seems to work for most C programs. It should eliminate a lot of time in the translation process.
Revision history:
1.00 Initial Release 4/29/87
Copyright 1987 by:
Dale W. Thompson, 14500 Dallas Pkwy. #2091, Dallas, TX 75240
This module and/or its procedures may be freely used by anyone, but please acknowledge its use in any copyright notice of a publicly distributed program. Thank you.
Please forward any comments, problems, or suggestions to me at the address given, or to my CompuServe ID 75115,734.
*)

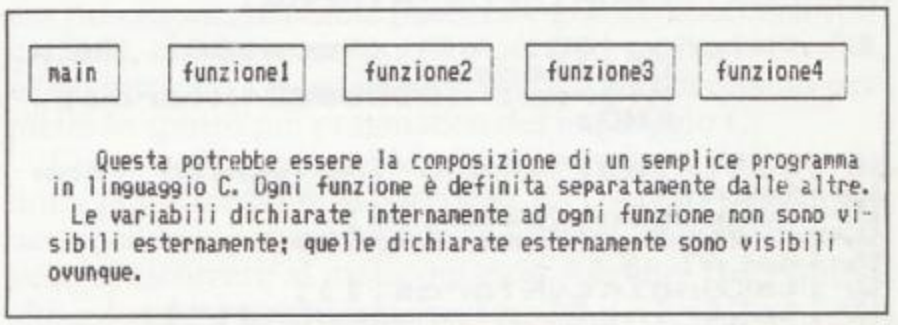
Inizio di un modulo di un programma che aiuta a convertire sorgenti dal linguaggio C al Modula 2.

Amiga BASIC ci muoveremo in un ambiente di lavoro semplice ed abbastanza vicino alla lingua naturale, ma raramente otterremo risultati professionali; viceversa con l'**assembler** potremo fare praticamente tutto, ma a prezzo di un impegno e di una fatica incomparabilmente maggiori. In genere si dice che l'*assembler* è a **basso livello**, in quanto è vicino all'*hardware* della macchina e che il Basic è ad **alto livello**, poiché è più vicino al linguaggio naturale. Esistono situazioni in cui il Basic e l'Assembler sono già la scelta migliore, tuttavia sorge spontanea una domanda: è possibile ottenere in qualche modo da un linguaggio di alto livello una potenza decisamente superiore a quella del BASIC, senza rinunciare ad una certa astrazione rispetto al linguaggio-macchina, ed a tutte le facilitazioni che essa comporta? Il tentativo di rispondere positivamente a questa domanda ha condotto alla creazione di diversi linguaggi che hanno permesso enormi passi avanti nell'informatica; i più noti fra questi linguaggi sono il **Pascal** (creato da N. Wirth e utilizzato ampiamente nell'MS-DOS, noto a partire dal 1970) ed il **linguaggio C** (creato da Kernigham e Richie, elemento portante del sistema UNIX nonché del nostro Amiga; è stato definito dagli stessi autori in un testo che è apparso per la prima volta nel 1978).

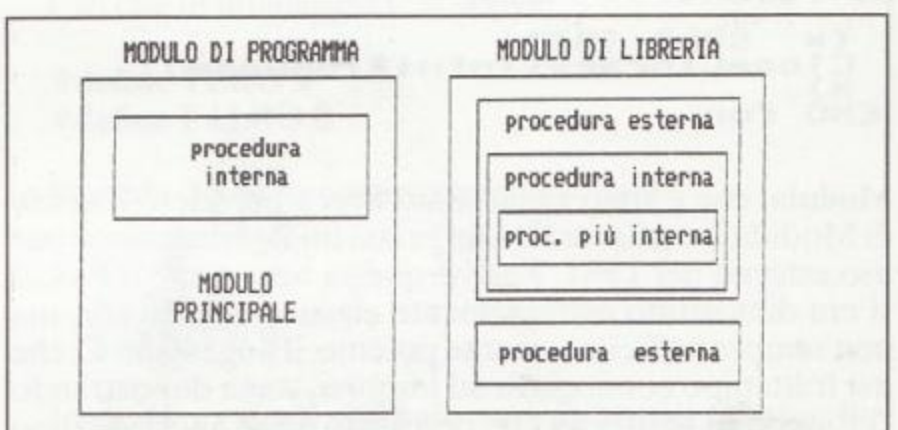
Mentre solitamente un programma in BASIC è solo un file di dati che deve essere eseguito istruzione per istruzione dal programma **interprete**, con grande dispendio di tempi d'esecuzione e di memoria, un programma in Pascal o in C è un file eseguibile indipendente, generato da un **compilatore**, che serve solo nel momento della produzione del codice. Ma questo *handicap* del BASIC standard può essere superato attraverso opportuni compilatori; in effetti il vero grande passo in avanti, che per primo il Pascal ha compiuto con decisione, è stato quello di permettere una **programmazione strutturata**.

LA PROGRAMMAZIONE STRUTTURATA

Alla programmazione strutturata abbiamo già accennato, e non è questa la sede per approfondire l'argomento; basti ricordare che essa permette sostanzialmente di:
1) controllare il flusso sequenziale di esecuzione del programma attraverso «cicli» (**loop**) di ogni tipo e che hanno precise condizioni di entrata e di uscita (chi programma in BASIC pensi, per esempio, al costrutto FOR ... NEXT); questi cicli permettono di strutturare il flusso in **blocchi**, in base a una logica intrinseca e autoesplicativa; si possono così eliminare i numeri di linea e porre al bando l'irrazionale istruzione GOTO;



Questa potrebbe essere la composizione di un semplice programma in linguaggio C. Ogni funzione è definita separatamente dalle altre. Le variabili dichiarate internamente ad ogni funzione non sono visibili esternamente; quelle dichiarate esternamente sono visibili ovunque.



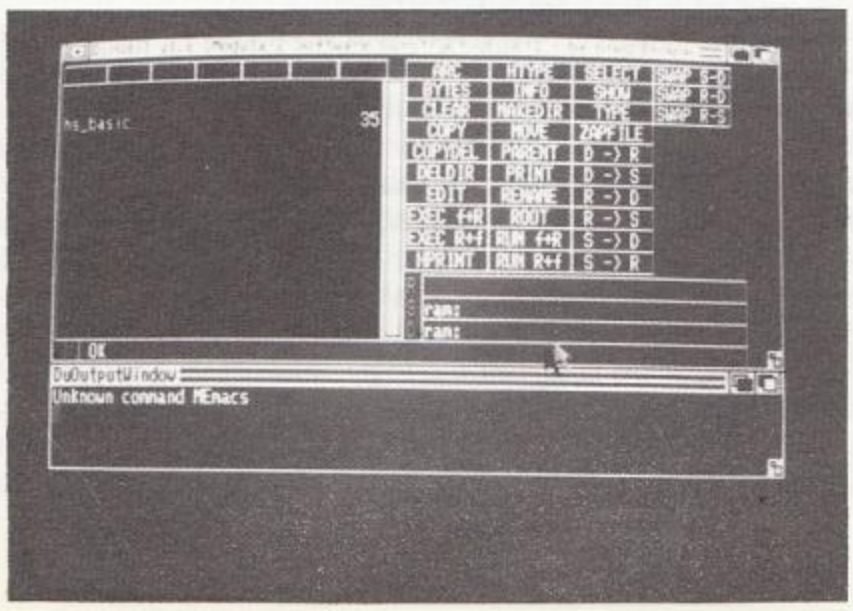
Questo schema rappresenta un semplice programma in Modula 2. Le procedure possono essere annidate. Le variabili sono visibili anche nelle procedure più interne rispetto a quella in cui sono dichiarate.

2) organizzare i dati in **record** (Pascal) o **strutture** (linguaggio C), così da poter far riferimento a gruppi di dati in modo coerente ed efficiente (questa possibilità è esclusa dal BASIC standard);
3) «spezzettare" il programma in parti indipendenti e sviluppabili separatamente (**procedure** in Pascal e **funzioni** in C).

La maggior potenza che i linguaggi ad alto livello hanno così ottenuto, quasi paradossalmente non significa un avvicinamento al linguaggio-macchina, ma un aumento dell'astrazione e della capacità di rendere visivamente percepibile, attraverso il listato, la logica di un programma. La costruzione di algoritmi complessi diventa così più intuitiva, mentre risulta anche enormemente facilitata la **lettura** del programma, vale a dire la possibilità di cogliere la funzione e il «senso» di un algoritmo attraverso l'esame del listato.

Nonostante la grande diffusione del Pascal, N. Wirth ha lavorato attorno a una nuova e più potente creatura: il

Un comodo programma di gestione file scritto in Modula 2 e «liberamente ispirato» ad un originale in C (DirUtil).



http://www.oldgamesitalia.net/
http://www.oldgamesitalia.net/

http://www.oldgamesitalia.net/

For.Mod

```
MODULE for;

(*
   test di un loop for 1-100.000, tempi medi:
   Benchmark M2: 1 secondo                (1 secondo,      12 microsecondi)
   TDI MODULA-2: 1 secondo                (1 secondo,      11 microsecondi)
*)

FROM SYSTEM                    IMPORT  ADR;
 (********** SOLO PER TDI **************
 FROM Libraries                 IMPORT OpenLibrary, CloseLibrary;
 FROM Intuition                 IMPORT IntuitionName, IntuitionBase;
 *)
FROM Intuition                 IMPORT CurrentTime;
FROM InOut                     IMPORT  WriteString, WriteLn;
FROM LongInOut                 IMPORT  WriteLongInt;

VAR
        Secs, EndSecs, Micros, EndMicros : LONGINT;
        i : LONGCARD;

BEGIN

 (*  SOLO TDI!
 IntuitionBase := OpenLibrary(IntuitionName,0);
 *)

CurrentTime(ADR(Secs),ADR(Micros));

FOR i:=0D TO 100000D DO
      END;

CurrentTime(ADR(EndSecs),ADR(EndMicros));

Secs:=EndSecs-Secs;
Micros:=EndMicros-Micros;

IF (Micros<0D) AND (Secs>0D) THEN
        DEC(Secs);
        Micros:=1000000D+Micros;
        END;

WriteString("tempo impiegato loop 1-100.000:");
WriteLn;
WriteLongInt(Secs,1);
WriteString(" secondi");
WriteLn;
WriteLongInt(Micros,1);
WriteString(" microsecondi");
WriteLn;

 (*  SOLO TDI!
 CloseLibrary(IntuitionBase);
 *)
END for.
```

Ecco un esempio di conversione da C a Modula 2: il programma è un test di velocità.

For.c

```
/*
   test di un loop for 1-100.000, tempi medi:
   Lattice: 6 decimi di secondo (0 secondi, 600007 microsecondi)
   Aztec:   1 secondo           (1 secondo,     12 microsecondi)
*/

#include (exec/types.h)
LONG    IntuitionBase;
main()
{
        LONG    secs, end_secs, micros, end_micros;
        int     i;
        IntuitionBase=(LONG)OpenLibrary("intuition.library",NULL);
        CurrentTime(&secs,&micros);
        for (i=0; i<100000; i++);
        CurrentTime(&end_secs,&end_micros);
        CloseLibrary(IntuitionBase);
        secs=end_secs-secs;
        micros=end_micros-micros;
        if (micros<0 && secs>0)
                {
                --secs;
                micros=1000000+micros;
                }

        printf ("tempo impiegato loop 1-100.000:\n %ld secondi\n
                 %ld microsecondi\n",secs,micros);
}
```

Modula, che è stato rielaborato fino a prendere il nome di Modula 2 e ad avere la sua prima implementazione per uso esterno nel 1981. Perché questa necessità? Il Pascal si era dimostrato estremamente elastico ed efficace, ma non sempre sufficientemente potente: il linguaggio C, che nel frattempo cominciava ad imporsi, stava dimostrando l'efficacia di istruzioni che potessero agire anche a basso livello; inoltre, se si credeva veramente nella programmazione strutturata, bisognava portare al limite le sue possibilità, cosa che il Pascal non poteva fare; infine, si stavano elaborando i concetti di **programmazione modulare** e di **ingegneria del software**, che ampliano l'idea di programmazione strutturata.

Lo zoccolo portante (o nocciolo che dir si voglia) del Modula 2 sta proprio nell'idea del modulo; i **blocchi** della programmazione strutturata possono articolarsi in unità più ampie che vengono definite **moduli**. Un programma sarà così composto da uno o più moduli, ed ogni modulo da una o più **procedure**. A questo modo si potrà impostare, realizzare e correggere ogni modulo separatamente e conservarlo nella nostra «libreria», in modo da poterlo utilizzare (in tutto o in parte) in qualsiasi programma che vorremo realizzare in futuro; se consideriamo che con questo metodo si può permettere a più persone di lavorare contemporaneamente ai diversi moduli di uno stesso progetto, e che per utilizzare un modulo basta sapere *cosa* fa senza dover sapere *come* opera, risulta chiaro come l'espressione «ingegneria del software» sia in questo senso appropriata.

Per questa e per altre particolarità il Modula 2 si propone quasi come un linguaggio «perfetto» o «ideale»; ma anche la perfezione si paga e, dalla misura in cui si paga, ognuno di noi può decidere se vale la pena di essere perfetti.

I MODULI E LE PROCEDURE

Esistono moduli-programma e moduli di libreria. I **moduli-programma** contengono l'intelaiatura-cardine del programma, la procedura principale, quella che in linguaggio C chiameremmo **main**; se il nostro programma si chiama, per esempio, *Editor*, questo modulo dovrà portare l'intestazione «MODULE Editor»; prima della procedura principale potranno comparire altre procedure, ed è evidente che esse saranno utilizzabili esclusivamente dal nostro programma.
I moduli di libreria conterranno invece le **procedure/funzioni** inseribili in qualsiasi progetto e che lavoreranno costantemente in coppia: ad ogni modulo-implementazione

http://www.oldgamesitalia.net/ http://www.oldgamesitalia.net/

http://www.oldgamesitalia.net/

corrisponderà sempre un modulo-definizione. Il **modulo-implementazione**, come il modulo-programma, è identificato dall'estensione «.mod» e deve portare l'intestazione «**IMPLEMENTATION MODULE ...**»; partendo da esso il compilatore genererà il codice-oggetto con le istruzioni in linguaggio-macchina che finiranno nel nostro codice eseguibile dopo il **linking**; è, insomma, il modulo che dice *come* fare. Il **modulo-definizione** dice invece *cosa* fare; porta l'estensione «.def» e l'intestazione «**DEFINITION MODULE ...**» ed il suo codice oggetto non è destinato a produrre codice eseguibile, ma a rendere possibile un ferreo e severissimo controllo da parte del compilatore sull'utilizzo delle funzioni ivi definite, con particolare riferimento al numero di parametri ed al loro tipo. Non pensiamo dunque di poter operare con la leggerezza che il C invece permette: questo fatto potrà crearci problemi e costringerci a conversioni di parametri, ma eviterà anche molti stupidi errori.

Le procedure possono o no restituire un valore, ed anche questo particolare è strettamente controllato. In C, per esempio, possiamo chiamare una funzione di qualsiasi tipo come se fosse *void*, cioè come se non restituisse alcun valore; in Modula 2 questo non è possibile. Un'altra differenza con il C è costituita dalla possibilità, offerta dal Modula 2, di *annidare* le procedure e da una gestione particolarmente elaborata delle regole di *visibilità*.

VISIBILITÀ, VARIABILI, COSTANTI

Al contrario che in BASIC, nei linguaggi più professionali come il C ed il Modula 2 è necessario dichiarare tutte le variabili definendone il nome e il tipo. Se, per esempio, dobbiamo utilizzare una variabile intera ed una a virgola mobile, in C dovremo scrivere qualcosa del genere:

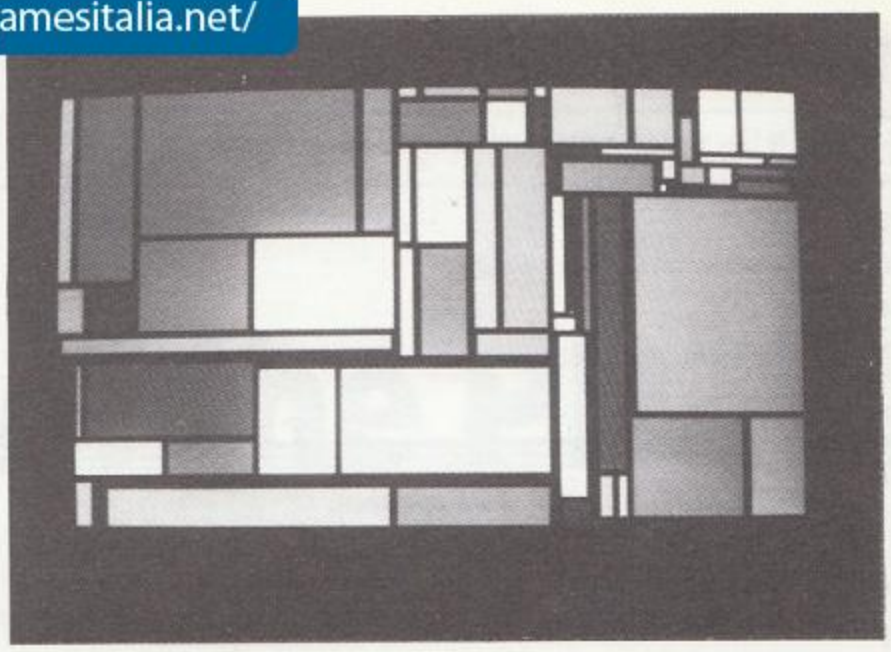
```
int intero;
float reale;
```

In Modula 2 la sintassi cambierà un po', ma non in modo radicale:

```
VAR
   intero : INTEGER;
   reale : REAL;
```

A parte la maggiore leggibilità, che avrete già notata, il Modula 2 offre, rispetto al C, un diverso spettro di utilizzo delle variabili all'interno del programma, caratteristica che prende il nome di «**visibilità**». Un programma in linguaggio C si presenta come una serie di funzioni ben separate fra loro; da una funzione si può utilizzare, o meglio si può «effettuare una chiamata» ad un'altra funzione, ma non è assolutamente possibile inserire dentro una funzione il codice di un'altra funzione; non è cioè possibile «**annidarle**». Questa possibilità viene invece offerta dal Modula 2, come è illustrato nel nostro schema.

In questo modo in linguaggio C abbiamo variabili globali, che sono dichiarate al di fuori di ogni funzione e risultano sempre «visibili» (cioè utilizzabili), oppure variabili locali, visibili solo dalla funzione che le ha dichiarate. In Modula 2, invece, ogni variabile è visibile solo alla procedura che la dichiara ed a quelle eventualmente definite al suo interno; a questo si aggiunga che una procedura più interna può utilizzare il nome di una variabile



Esempio di output di un programma Modula 2 che emula i quadri del noto pittore astrattista olandese Piet Mondrian.

esterna per dichiarare una propria, creando così una nuova variabile e rendendo automaticamente invisibile quella esterna. Questa particolarità può render bene lo spirito del linguaggio di cui ci stiamo occupando: massima precisione, massima potenza e grande coerenza concettuale, che possono portare però ad un flusso di controllo intricato e poco lineare, al contrario di quanto permette lo spirito più pragmatico del linguaggio C.

Come nel Pascal, nel Modula 2 possiamo definire dei nomi per le costanti; al contrario, il C — nella sua versione originaria — non prevede le costanti, consentendo invece di ricorrere al metacomando **#define**, con risultati che solo apparentemente sono uguali.

Ciò che in linguaggio C si scrive:

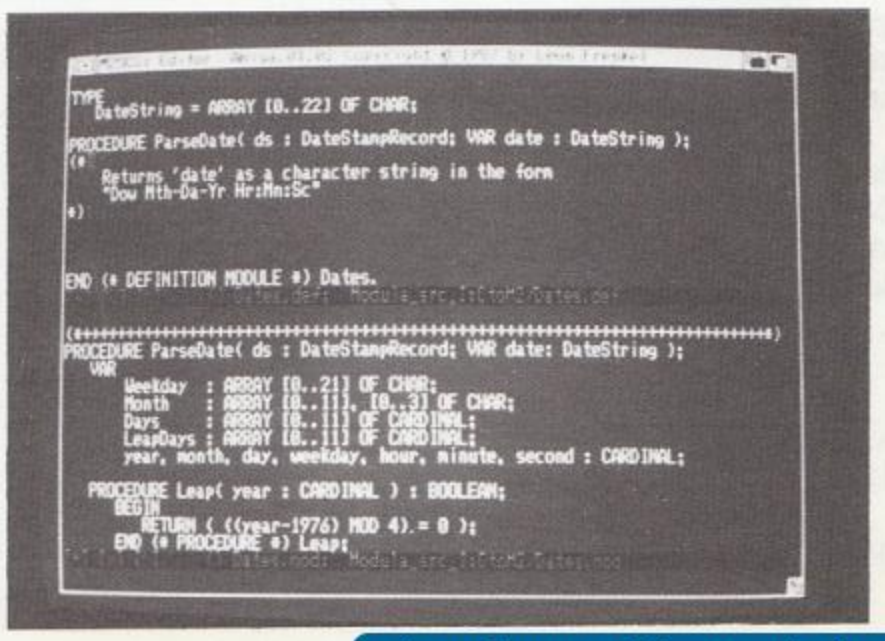
```
#define VERO 1
#define FALSO 0
```

in Modula 2 può apparire come:

```
CONST
   VERO = 1;
   FALSO = 0;
```

In questo caso il risultato sarà lo stesso nei due linguaggi; tuttavia il #define del C, a differenza del tipo CONST, è in realtà quella che in linguaggio *assembler* si chiama MACROISTRUZIONE: al momento della lettura del sorgente l'espressione VERO viene sostituita con 1 e FALSO con 0. Questo è uno dei pochi veri limiti del

L'editor del «Benchmark Modula 2» è a finestre multiple. Qui sono visualizzati contemporaneamente un modulo definizione ed il corrispondente modulo implementazione.



```
TYPE
   DateString = ARRAY [0..22] OF CHAR;

PROCEDURE ParseDate( ds : DateStampRecord; VAR date : DateString );
(*
   Returns 'date' as a character string in the form
   "Dow Mth-Da-Yr Hr:Mn:Sc"
*)


END (* DEFINITION MODULE *) Dates.
```

```
(*+++++++
```

http://www.oldgamesitalia.net/

```
hello.mod
MODULE Hello;

(*
  manda "Hello, world!" verso lo standard-output.
  Dimensioni eseguibile:
  Benchmark M2:   7612 bytes;
  TDI Modula-2:  15992 bytes;
*)

FROM InOut IMPORT WriteString, WriteLn;

BEGIN

WriteString("Hello, world!");
WriteLn;

END Hello.
```

```
hello.c
/*
  manda "Hello, world!" verso lo standard-output.
  Dimensioni eseguibile:
  Lattice:   6376 bytes;
  Aztec:     5912 bytes
*/

main()
{
printf("Hello, world!\n");
}
```

Il classico programma dimostrativo che visualizza sullo schermo un messaggio, in versione Modula 2 e in versione C.

Modula 2, poiché la macro risulta in effetti molto potente; se in linguaggio C scrivo:

#define dispari(var) (var & 1)

otterrò la possibilità di usare l'istruzione *dispari()* come fosse una funzione, mentre in realtà sarà un'espressione di tipo booleano. Col Modula 2 tutto ciò non è affatto possibile: si dovrà scrivere una procedura vera e propria, con un incremento dei tempi di stesura e di esecuzione del programma.

LEGGIBILITÀ E PROCEDURE DI COMPILAZIONE

È naturalmente impossibile dare qui una descrizione completa delle caratteristiche del Modula 2: come si è detto, questo linguaggio privilegia la coerenza logico-strutturale, il rigore nel controllo dei tipi, la programmazione modulare e la «leggibilità» del codice sorgente. Già il Pascal era molto leggibile; in questo senso (e in contrasto con il linguaggio C) sono stati fatti numerosi passi avanti. Il Modula 2 favorisce, per sua natura, nomi di variabili piuttosto lunghi ed autoesplicativi; questi nomi devono essere composti da caratteri alfanumerici ed iniziare con un carattere alfabetico; il «Benchmark M2» è particolarmente favorevole in questo senso, in quanto consente fino a 255 caratteri e permette (cosa non prevista dallo standard) il carattere «_» (**underscore**), tanto utilizzato nel C.

La leggibilità è anche favorita dal forzato utilizzo delle parole-chiave in caratteri maiuscoli, come accade normalmente nel BASIC, e dal modo efficace con cui sono implementati i **loop** e le condizioni (IF..., CASE...).

Fondamentale per il concetto di modularità è la parola-chiave **IMPORT**, che è praticamente esclusiva del Modula 2. Con essa si «importa» un oggetto (una procedura, una funzione, una costante, ecc.) da un modulo di libreria. Se, per esempio, scrivo:

FROM InOut IMPORT WriteString, WriteLn;

significa che nel modulo di **libreria InOut** (questo modulo viene fornito da ogni compilatore M2, ma, come si è visto, anche noi possiamo realizzare moduli di libreria) potrò utilizzare o «vedere» le procedure **WriteString** (che scrive una stringa) e **WriteLn** (che manda a capo il cursore) e solo quelle! Quando il compilatore leggerà questa istruzione, si preoccuperà di verificare l'esistenza dei due *file* di libreria e in particolare utilizzerà il codice oggetto prodotto dal sorgente **InOut.def** per verificare la correttezza della nostra sintassi (e darci una bacchettata sulle dita in caso di errore). Il **Linker** invece utilizzerà questa istruzione per prelevare dal giusto modulo di libreria il codice-macchina, relativo alla procedura che intendiamo importare, e che verrà inserito nel *file* eseguibile. Questa gestione «controllata» delle importazioni spiega perché il *Linker* dei compilatori M2 non è mai compatibile con il nostro beniamino *BLink* e perché risulti nel contempo molto più facile da lanciare (in quanto non bisogna specificare codici oggetto né *file* di libreria). D'altra parte questo rende spesso problematico l'inserimento di procedure scritte con altri linguaggi in un programma in Modula 2.

IL MODULA 2 E IL C

Come era inevitabile, siamo entrati nel difficile terreno del confronto fra il Modula 2 ed il C. In effetti esistono molti altri linguaggi, fra cui alcuni particolarmente interessanti, ma nel campo della programmazione tradizionale su piccoli sistemi questi due linguaggi sono in realtà le due possibilità più convincenti, specie su di una macchina come Amiga, che non offre valide implementazioni del Pascal. Tutto questo ha purtroppo comportato la formazione di opposti schieramenti e, spesso, anche di polemiche. Un vero confronto tra C e Modula 2 non è cosa semplice: esistono libri su questo argomento. Noi ci siamo limitati ad indicare le caratteristiche generali del Modula 2 ed a fare un accenno alle differenze rispetto agli altri linguaggi.

Grandi consigli non vogliamo né, in effetti, possiamo darne: se i vostri obiettivi sono limitati o specifici, scegliete il BASIC (senza escludere Amos), altrimenti puntate sul Modula 2 o sul C, a seconda delle vostre preferenze, ricordando comunque che una discreta conoscenza dell'*assembler* non vi nuocerà mai, se non altro per inserire nei vostri programmi in BASIC, C o M2 delle procedure ottimizzate per efficienza e velocità d'esecuzione.

UNA BUONA SCELTA

Concludiamo dicendo che una buona parte della bontà di un compilatore dipende anche dalla specifica implementazione del linguaggio: il «Benchmark M2» ci è sembrata una buona scelta, perché permette un ambiente integrato, vale a dire la possibilità di scrivere il listato e, dallo stesso ambiente, compilare il programma e lanciarlo per verificarlo. Abbiamo già apprezzato queste caratteristiche nell'**HiSoft Basic**, notando come ciò permetta di avvicinarsi alla praticità di un linguaggio interpretato. Nel nostro caso abbiamo anche un *debugger* di sorgente, che offre il grande vantaggio di eseguire il nostro codice riga per riga, proprio come il *trace mode* del BASIC. A risentirci alla prossima volta, per esaminare nei dettagli le funzionalità di questo compilatore.

http://www.oldgamesitalia.net/ http://www.oldgamesitalia.net/

http://www.oldgamesitalia.net/

The best of the PD

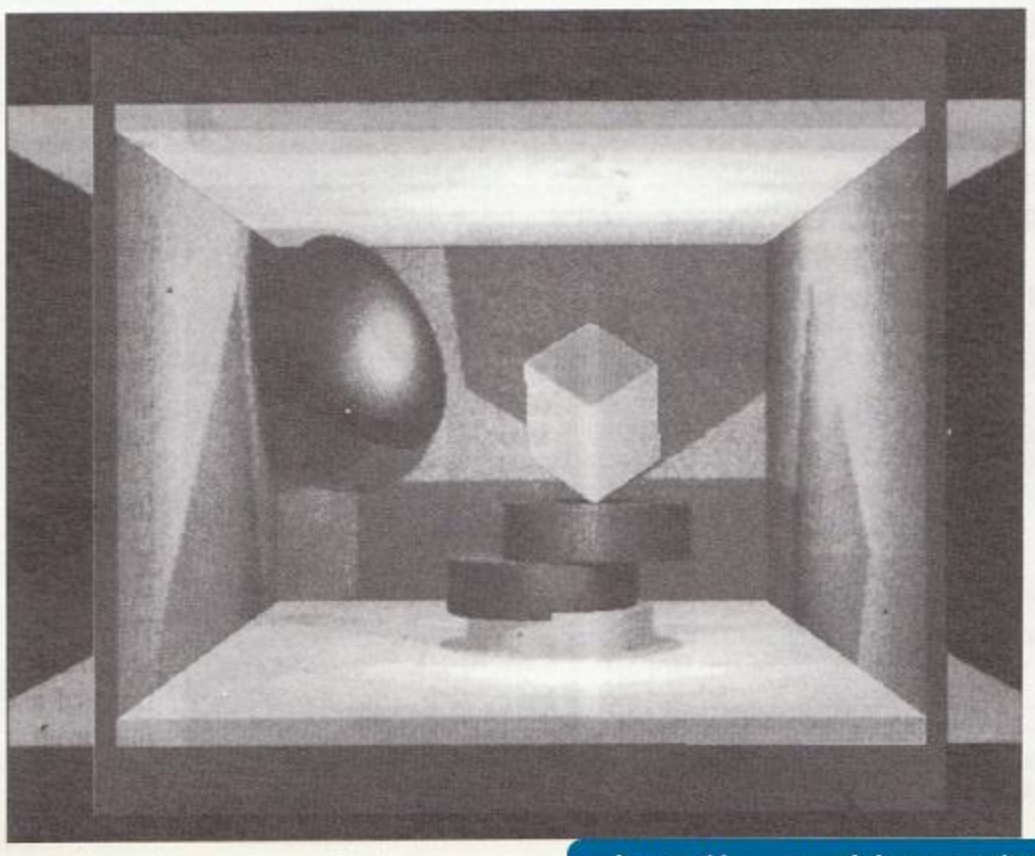
Abbiamo scelto per voi, e continueremo a farlo, i migliori programmi fra quelli di Pubblico Dominio esistenti che, per ragioni di spazio, non possono essere inseriti nel dischetto allegato ad Amiga Byte.

di EMANUELE SCRIBANTI

C-LIGHT 1.03

Originariamente distribuito come commerciale al prezzo di circa 50 dollari, il programma di ray tracing «C-Light» è diventato ora di pubblico dominio/shareware.
Pur non essendo ovviamente in grado di competere sul piano delle prestazioni con pacchetti come «Turbo Silver» o «Sculpt Animate 4D», «C-Light» è comunque particolarmente indicato per chi muove i primi passi nel settore della grafica ray tracing con Amiga, essendo veramente molto facile da usare. Per questo motivo, nonostante la mancanza del supporto del modo HAM e la discreta lentezza, «C-Light» rappresenta la scelta ideale per il principiante poiché, a differenza di altri programmi PD di ray tracing come «DKBTrace» o «QRT», è dotato di un editor grafico di oggetti utilizzabile tramite il mouse, e non richiede lunghe e complesse digitazioni di file ascii per la definizione degli oggetti e della scena.
I lettori più assidui ricorderanno che AmigaByte descrisse il programma nelle pagine del fascicolo 15; a beneficio di chi avesse perso quel numero, riassumiamo le caratteristiche salienti di «C-Light» riprendendo parzialmente il testo di quella recensione.
I due corpi principali del programma sono «l'Object Editor» ed il «Ray Tracer». Con il primo creeremo oggetti tridimensionali servendoci di un'unica finestra di lavoro, grazie ad alcune semplici primitive geometriche quali la sfera ed il cilindro. Pochi ma efficaci «strumenti» di lavoro consentono di posizionare e manipolare gli oggetti. Completato l'editing, il contenuto della scena viene salvato in un file Ascii e si passa al secondo programma, quello per la gestione del rendering. Il Ray Tracer carica il file precedentemente creato con l'Object Editor e calcola l'immagine finale; viene supportato, oltre che il modo overscan, lo standard europeo PAL, con o senza interlace. Data la semplicità del software, C-Light può girare tranquillamente su qualsiasi Amiga privo di espansione.
Lasciando «C-LightI» si accede all'editor tridimensionale, il cui unico menu a tendina è semplicissimo, anche se un po' scarno. È bene ricordare che, qualora si ritenga di utilizzare un disco «Data:» predisposto per la registrazione dei lavori, bisognerà informare il programma con il comando «New Directory», perché il menu di I/O non consente nessun tipo di indirizzamento dei file. Sul lato destro dello schermo di lavoro troviamo i comandi di editing atti a realizzare i solidi. I primi tre gadget permettono la creazione istantanea delle tre primitive grafiche disponibili: la sfera, il cilindro ed il cubo; basterà attivare uno di questi tre gadget perché il programma crei il solido selezionato disegnandolo in modalità «Wire Frame». Si potrà spostare l'oggetto nello spazio con il mouse: muovendo semplicemente quest'ultimo, l'oggetto si sposterà sul piano parallelo allo schermo (alto/basso e destra/sinistra); agendo sul tasto destro potremo posizionare il solido sul piano perpendicolare al monitor (in/out). Poiché lo spazio virtuale di «C-Light» è notevolmente più ampio dello schermo di lavoro, l'oggetto non segue perfettamente il puntatore del mouse.
Il posizionamento degli oggetti all'interno dell'universo di «C-Light» è più facile usando le coordinate che si trovano sulla barra superiore dello schermo, la quale indica la posizione assoluta del centro dell'oggetto rispetto all'origine. Posizionato il solido, si clicka il tasto sinistro del mouse e si accede ad un requester denominato «Object Modification Requester». Gli slider posti orizzontalmente consentono di dimensionare l'oggetto selezionato lungo i tre assi fondamentali X, Y e Z.
Altri tre slider rendono possibile, in maniera analoga la rotazione del solido. Con i gadget A, B, C ed M si seleziona il colore delle

superfici della figura selezionata; il gadget M imposta una particolare caratteristica ottica del solido e lo rende «Specular» (le superfici interessate riflettono totalmente i raggi provenienti dalla sorgente luminosa). Mantenere il requester in funzione e, contemporaneamente, utilizzare il mouse per spostare il solido nello spazio per ottenere posizioni e dimensioni ottimali è decisamente comodo. Per cancellare l'oggetto si attiva il gadget «Cancel»; un click sul «Close Gadget» fa uscire dal requester.
La prima fase dell'editing è conclusa. A questo punto tornano molto utili i due comandi posti al di sotto degli «Add Object Gadget»; il classico «Delete» ed il comando «Change» (CHG), se si vogliono apportare modifiche al solido selezionato.



http://www.oldgamesitalia.net/

http://www.oldgamesitalia.net/

http://www.oldgamesitalia.net/

AMIGABYTE È LA TUA RIVISTA !

Contribuisci anche tu a renderla sempre più bella ed interessante rispondendo a qualche domanda ed aiutandoci a darti ciò che vuoi!

Quanti anni hai? .....

Di cosa ti occupi (studio, lavoro)?.....

.....

Quale Amiga possiedi?

☐ 500 ☐ 1000

☐ 2000 ☐ 3000

Quali di queste periferiche possiedi?

☐ stampante ☐ disk drive esterno ☐ hard disk

☐ modem ☐ espansione di memoria

☐ schede acceleratrici

☐ genlock o digitalizzatore video

☐ equipaggiamento midi o digitalizzatore audio

☐ schede emulazione MsDos o SideCar

☐ altro

Da quanto leggi AmigaByte? .....

.....

La trovi facilmente in edicola?.....

.....

Di AmigaByte segui di più:

☐ la rivista

☐ il dischetto

☐ entrambi

Sei più interessato a:

☐ recensioni utility ☐ recensioni giochi

☐ didattica ☐ altro

Quali temi preferisci?

☐ musica ☐ grafica e animazione

☐ programmazione (Basic, C, Assembler)

☐ applicativi (word processing, database, spreadsheet)

☐ giochi e avventure ☐ desktop publishing

☐ desktop video ☐ prove hardware

Quali sono le tue rubriche preferite di AmigaByte?

☐ Software Express ☐ Tips & Tricks

☐ Adventure ☐ I corsi di programmazione

☐ Tools ☐ Le recensioni/tutorial

☐ Megagame ☐ Altro

Che livello di approfondimento desideri per gli articoli?

☐ Per principianti ☐ Intermedio ☐ Per esperti

Che voto daresti in media ad AmigaByte?

Disco:	Pessimo	Rivista:	Pessima
	Scarso		Scarsa
	Discreto		Discreta
	Buono		Buona
	Molto buono		Molto buona
	Eccellente		Eccellente

Ti piacciono le copertine di AmigaByte?.

.....

Cosa vorresti trovare sulla rivista? .....

.....

Cosa vorresti trovare sul disco? .....

.....

Quali altre riviste leggi? .....

.....

Hai dei suggerimenti? Quali? .....

.....

Nome ..... Cognome .....

Indirizzo .....

Città .... Prov..... C.A.P. .....

Inserisci i tuoi dati (facoltativi) e spedisci questo tagliando, o una sua fotocopia, ad AMIGABYTE, C.so Vittorio Emanuele 15, 20122 Milano.

Si può modificare l'intera scena con il comando «View», che disegna un cubo al posto degli oggetti creati ed attiva l'«Object Modification Requester», mediante il quale possiamo modificare la dimensione di tutti gli oggetti contemporaneamente, ma anche la dimensione di ognuno e l'orientamento.
Con il comando «Eye» è possibile determinare la distanza dell'osservatore dalla scena: il gadget attiva il solito requester, che permette esclusivamente di spostare il punto di vista in un'unica direzione, perpendicolare al piano individuato dalla scena.
Per posizionare le sorgenti luminose si attiva il gadget «Lite»: la fonte di luce viene considerata come un qualsiasi oggetto, quindi può essere posizionata con il mouse come descritto più sopra. «C-Light» può gestire fino ad un massimo di 170 sorgenti di luce distinte: naturalmente la presenza di un elevato numero di luci aumenta notevolmente i tempi di calcolo.
Le tre differenti tinte accessibili dai gadget A, B e C dell'«Object Modification Requester» s'impostano attivando il comando «Color», che apre un particolare screen con gli slider relativi ai tre colori fondamentali RGB. Il colore attribuito alla superfice verrà assegnato ai punti più vicini alla sorgente luminosa; il colore andrà «sfumando» verso tinte più scure in modo graduale via via che ci si allontana dalla sorgente stessa.
«C-Light II», il secondo modulo di «C-Light», effettua il ray tracing vero e proprio, e va lanciato dopo la realizzazione dell'object file mediante «C-Light I».
«C-Light II» trasforma i numeri e le definizioni del «file oggetto» in una scena tridimensionale realistica. Per completare una singola immagine occorre un tempo che oscilla dai pochi minuti, per scene particolarmente semplici, fino a diverse ore, necessarie per «renderizzare» un file complesso e ricco di sorgenti luminose.
Un'unica finestra contiene i comandi essenziali per un corretto rendering. Con i tre slider superiori si controllano l'intensità luminosa delle luci (Light), il livello di luce di «background», che illumina gli oggetti indipendentemente dalla loro posizione (Ambient), ed il livello di «blending» che controlla l'omogeneità delle sfumature di colore (Blend).
Nella parte inferiore della finestra di lavoro sono poi presenti alcuni gadget che attivano l'antialiasing (algoritmo che riduce il livello di demarcazione tra due colori differenti lungo una linea, attenuando il cosiddetto effetto «scaletta»), il calcolo delle ombre ed i requester di Input/Output. Per lanciare la fase conclusiva si clicka sul gadget «GO!» e si attende con pazienza il termine delle operazioni.

Disponibile sul dischetto: AmigaByte PD 70

Per ricevere i dischetti contenenti i programmi recensiti su queste pagine, basta inviare vaglia postale ordinario di Lire 10 mila ad AmigaByte, c.so Vitt. Emanuele 15, 20122 Milano, specificando il codice del disco (esempio: Amiga Byte PD 2) ed i propri dati chiari e completi.

http://www.oldgamesitalia.net/

http://www.oldgamesitalia.net/

http://www.oldgamesitalia.net/

HEAP ED ICONE

Ho da poco acquistato un Amiga 500. Per ora mi limito a fare qualche semplice programma con AmigaBasic e gradirei la vostra consulenza su un paio di cose.

Programmando in Basic a volte mi appare la scritta «Out of Heap Space» racchiusa in una cornice rossa, ma non riesco a spiegarmi il perché e soprattutto a trovare un rimedio.

Inoltre: come faccio a far apparire sullo schermo l'icona di un mio disco di lavoro e, clickandoci sopra, a farne apparire altre, magari racchiuse all'interno di una finestra nella posizione desiderata?

Ermanno Manzoni - Pioltello

AmigaBasic suddivide la memoria in tre differenti categorie: il Data Segment, lo Stack ed il System Heap. Il Data Segment rappresenta la quantità di memoria riservata al listato Basic, ed ammonta generalmente di 25000 byte. Lo Stack è la zona di memoria riservata alla memorizzazione temporanea delle variabili interne di AmigaBasic, e durante l'esecuzione di un programma contiene informazioni che comunicano all'interprete il numero di linea al quale tornare dopo un GOSUB, i valori di un ciclo FOR ... NEXT e così via. La restante quantità di memoria presente nel sistema viene usata per l'allocazione dei dati (ad esempio schermi e finestre o immagini IFF caricate in Ram) sfruttati da comandi come Window, Screen, Sound, Wave, etc.

Il comando «Out of Heap Space» indica che questo spazio si è esaurito ed equivale, in pratica, ad un avviso di memoria insufficiente per l'esecuzione del programma. Occorre perciò controllare che il programma non tenti di allocare memoria in maniera errata, ad esempio aprendo schermi con risoluzione troppo elevata, o tentando di gestire un numero eccessivo di finestre allo stesso tempo. La soluzione migliore in questi casi consiste nel dotare il computer di un'espansione di memoria. Le dimensioni dell'Heap possono essere visualizzate con la funzione FRE(−1).

Per quanto riguarda le icone, esse sono memorizzate su di un dischetto sotto forma di file aventi lo stesso nome del programma o del cassetto cui si riferiscono, più l'estensione «.info», sempre in minuscolo. Ad esempio, l'icona di un programma o di un cassetto chiamato «Utility» si chiamerà «Utility.info». L'icona di un dischetto, quella che appare inserendo il disco in ambiente WorkBench, ha invece un nome convenzionale fisso: «Disk.info». Per attribuire un'icona

ad un disco, è sufficiente copiarci il file «Disk.info» preso da un altro dischetto, o crearlo con un editor di icone.

La posizione delle icone in una finestra e le dimensioni di quest'ultima possono essere fissate con il comando «SnapShot» del menu «Special» del WorkBench. Occorre posizionare le icone e la finestra secondo le proprie esigenze, tenere premuto il tasto Shift e clickare una sola volta su tutte le icone desiderate, per selezionarle. Dopo averle attivate, è sufficiente richiamare da menu l'opzione SnapShot con il tasto destro del mouse ed attendere che le loro coordinate siano memorizzate sul dischetto.

PROBLEMI CON SMOVIE

Ho tentato inutilmente di caricare il programma «SMovie», incluso nel dischetto di AmigaByte 28, ma non sono riuscito a cavare un ragno dal buco: quando clicko sul riquadro contenente il nome del programma, mi appare una richiesta di inserire il nome del file da visualizzare e a questo punto non so che fare.

I testi dimostrativi forniti con il programma invece si caricano senza problemi: ma come faccio a realizzare titoli personalizzati?

Ugo Parolini - San Donato Milanese

Come spiegato, forse un po' sommariamente, nella documentazione allegata al programma, «SMovie» visualizza il contenuto dei file di testo e lo fa scorrere sullo schermo come i titoli di un film. Pertanto occorre preparare preventivamente il testo mediante un qualsiasi editor e memorizzarlo su d'un dischetto sotto forma di file ascii.

Il metodo più semplice consiste nell'uso dell'editor «Ed», presente su ogni dischetto WorkBench. Caricate normalmente il disco WorkBench, entrate in Cli e digitate ad esempio
Ed Disco:NomeFile
dove «disco» è il nome di un dischetto precedentemente formattato e «nomefile» è il nome con il quale deve essere salvato il file di testo. Una volta terminato l'inserimento del testo, potete salvarlo digitando i tasti «Esc» e poi «X» seguito da Return.

A questo punto, quando «SMovie» vi chiederà il nome del file da visualizzare, potrete rispondere «Disco: Nomefile». Il programma richiederà l'inserimento del dischetto avente il nome specificato, e visualizzerà correttamente sullo schermo il testo da voi digitato.

AMIGA BYTE LINE - 02/79.50.47

La Redazione risponde il mercoledì pomeriggio
dalle 15 alle 18 a voce alle vostre telefonate.

BBS 2000 - 02/76.00.68.57 - 24 ore su 24

La Redazione risponde via modem nell'area
«Linea diretta con AmigaByte». Collegatevi 300-1200-2400-9600-19200 baud.

RISERVATA AI LETTORI DI AMIGA BYTE

HOT LINE!

http://www.oldgamesitalia.net/

http://www.oldgamesitalia.net/

Scala... reale!

Dalla Norvegia, un programma per la creazione di titoli e presentazioni grafiche in grado di superare facilmente le prestazioni dell'agguerrita concorrenza «made in Usa»

di LUCA MIRABELLI

Per anni siamo stati abituati a software made in USA: dai simulatori di volo agli spreadsheet, tutto ciò che serviva per far funzionare il nostro computer proveniva da oltreoceano. Nella cara vecchia Europa, solo dall'Inghilterra proveniva ogni tanto qualche programma di qualità, soprattutto giochi. In Germania c'erano (e ci sono tuttora) gli hacker più spregiudicati e, se vogliamo, più abili, ma la loro produzione si limita alle demo, ed a qualche sporadica utility. Negli ultimi tempi, tuttavia, la produzione e la circolazione di software continentale sono aumentate in maniera considerevole: persino l'Italia, da sempre buona ultima nella produzione di software commerciale, sta trovando un suo spazio nel mercato, ed il talento dei programmatori europei comincia finalmente ad emergere.

Questo mese parliamo di «**Scala**», opera di un gruppo di programmatori e grafici norvegesi; i loro nomi sono visibili in una delle illustrazioni: riportarli nel testo avrebbe provocato una crisi di nervi ai linotipisti!

Commercializzato dalla Digital Vision a/s (Waldemar Thranesgt. 77, N0175 Oslo 1, Norvegia), «Scala» è una delle opere più notevoli che ci sia mai capitato di conoscere: il perché di questa affermazione, che può sembrare un tantino azzardata, sarà chiaro più avanti.

Per evitare illusioni inutili, è bene dire subito che, salvo modifiche, «Scala» dà il meglio di sè con un Megabyte di Chip RAM, possibilmente accompagnata da un po' di Fast.

Per farla breve, si tratta di un programma per slide show e titolazioni animate, realizzato con tanta cura dei particolari ed abiltità tecnica da far pensare che presto possa soppiantare «**Pro-Video**», che rimane tuttora il software più usato negli studi di produzione.

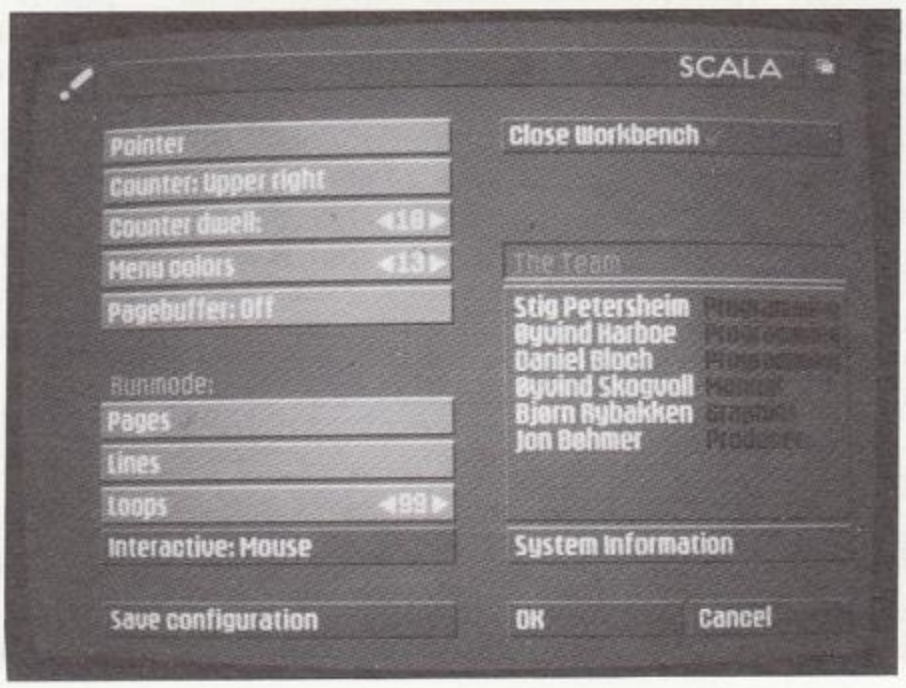
I PRIMI... SCALINI

Il pacchetto «Scala» è costituito da numerosi dischi: il primo contiene il programma principale, l'utility **ScalaPlayer** (che si occupa della presentazione degli slide show senza bisogno di caricare il programma principale), ed il programma che si incarica dell'eventuale installazione su hard disk; il secondo disco contiene i font, ed i rimanenti cinque dischi alcune librerie di sfondi.

Dopo il caricamento compare una pagina di storyboard, per il momento vuota. Una presentazione «Scala» è costituita da pagine (**Pages**), che a loro volta contengono uno sfondo (**Background**) e degli oggetti: testi (**Text**) o disegni (**Symbol**).

Ciascuna riga dello storyboard rappresenta una pagina della dimostrazione. Poiché per il momento sul dischetto non sono disponibili presentazioni dimostrative, vediamo in dettaglio come costruirne una partendo da zero.

Precisiamo subito che il dialogo tra utente e programma non avviene tramite i consueti menu a tendina, ma con pannelli di controllo. Questa soluzione si rivela molto efficace ai fini della velocità operativa: tutti i comandi sono sotto gli occhi, senza bisogno di andarli a cercare fra i menu. Per chi lo trovasse



Il menu System, per mezzo del quale ogni utente può personalizzare l'ambiente di lavoro.

http://www.oldgamesitalia.net/

http://www.oldgamesitalia.net/

http://www.oldgamesitalia.net/

più comodo, sono presenti le combinazioni «Amiga + tasto» corrispondenti a quasi tutte le funzioni del programma.

Per prima cosa selezioneremo **New**, per comunicare al programma che abbiamo intenzione di aggiungere una pagina. Sullo schermo comparirà un file requester, per mezzo del quale sceglieremo l'immagine di sfondo per questa pagina.

Già nel file requester troviamo una caratteristica assolutamente nuova e degna di nota: possiamo associare un pulsante alla directory attualmente visualizzata in modo che, per ritornarci, non dovremo indicarne ogni volta il path completo. Questa soluzione, che si colloca a metà tra i file requester più poveri (come quello di «Deluxe Paint») e quello, potentissimo ma a volte troppo confusionario, del Kickstart 2.0, farà la gioia dei possessori di hard disk.

Selezionando **OK** senza indicare alcun nome, lavoreremo su di uno sfondo neutro, del quale ci saranno richiesti i parametri (colore, risoluzione...).

Gli sfondi del programma lasciano a bocca aperta per la qualità. Realizzati in alta risoluzione overscan interlacciata, sono totalmente privi di sfarfallio grazie alla sapiente scelta dei colori; rappresentano in maggioranza delle superfici ruvide, e sono adattissimi per qualsiasi utilizzo. Imprimeteveli nella memoria: siamo pronti a scommettere che molto presto li vedrete in giro.

Lo storyboard della nostra presentazione: per ogni pagina sono indicate transizione e permanenza.

EDITING DELLE PAGINE

Scelta l'immagine, essa sarà caricata e visualizzata, con un cursore in alto a sinistra. Possiamo quindi cominciare a scrivere i testi associati a questa immagine. Le caratteristiche di queste linee di testo sono controllate dai gadget nella parte bassa. Vediamoli uno ad uno, a cominciare da quelli della prima riga.

Front serve per selezionare il colore principale del testo; ciò avviene selezionando un colore dalla palette, e successivamente il rettangolino nella parte destra del gadget.

Shadow aggiunge un'ombra al testo, e permette di sceglierne il colore. Solitamente, sia la bordatura che l'ombra sono nere.

3D rende il testo tridimensionale, mediante un processo di estrusione che dovrebbe essere familiare agli utenti di «TV*Text» o di «Title Page». Per tale processo sono necessari due colori, da scegliersi con le solite modalità.

MAKE-UP DEL TESTO

I sei gadget quadrati selezionano l'allineamento (a sinistra, a destra, centrale) ed il tipo di carattere (sottolineato, grassetto, corsivo). Il gadget con la scritta **Layout** permette di accedere al **Layout Editor**, uno degli aspetti più raffinati ed interessanti del programma.

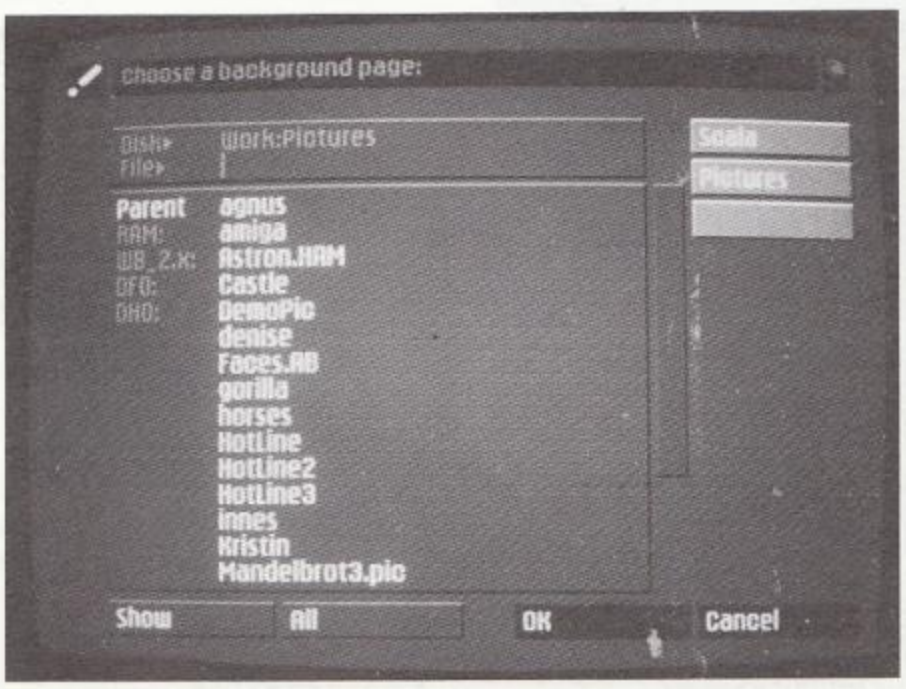
Con i nove pulsanti sulla sinistra si indica la direzione dell'ombra o dell'estrusione (a seconda dello stato del gadget sottostante). A dire il vero, l'utilità del gadget centrale è assai dubbia: il suo effetto è la moltiplicazione di un testo in colori e posizioni differenti, così da renderlo talvolta illeggibile. Molto probabilmente questa funzione deve ancora essere messa a punto.

Nella parte centrale del riquadro possiamo modificare i valori di nove parametri: ne sono visibili quattro alla volta, e per scorrerli tutti è necessario muovere verso il basso la scrollbar immediatamente alla loro sinistra.

I parametri sono: **Line spacing** (spaziatura tra le linee), **Char spacing** (spaziatura tra i caratteri), **Shadow Length** (distanza dell'ombra del testo), **Outline Size** (spessore della bordatura), **Italics** (inclinazione dei caratteri in corsivo), **Net Size** (dimensione della griglia: la incontreremo più avanti), **3D Length** (la dimensione dell'estrusione), **Bold Size** (la differenza tra le larghezze dei caratteri in grassetto e non), **Underline Size** (lo spessore della sottolineatura).

Alcune di queste finezze possono sembrare inutili, ma chi fa videografica sa bene quanto possano far comodo. In particolare, **Underline Size** permette finalmente di avere una sottolineatura proporzionata al corpo del carattere, e non di spessore fisso o addirittura di un pixel soltanto (che causa spaventosi sfarfallii in interlace).

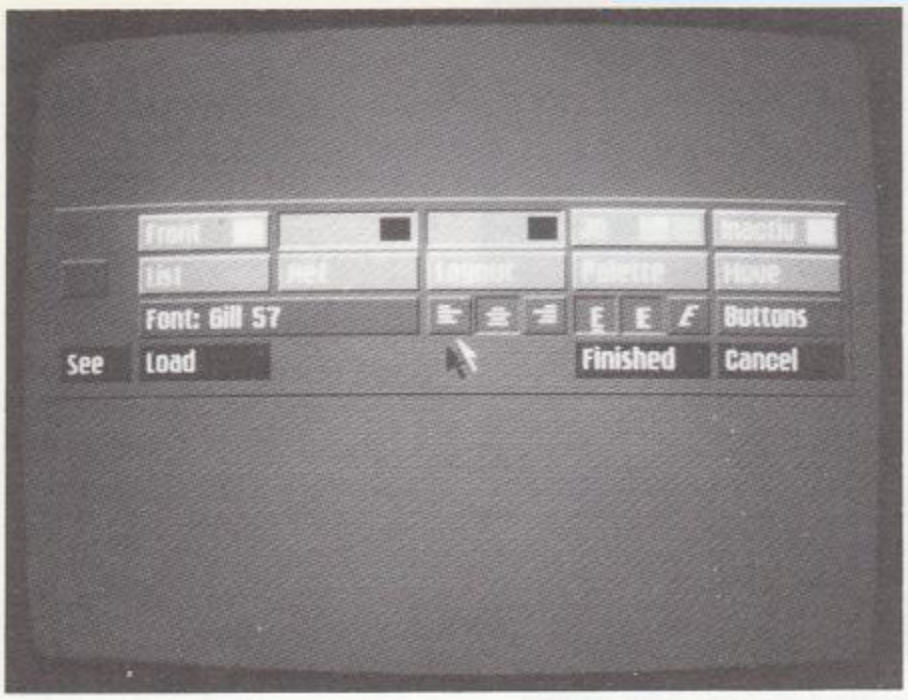
I triangoli nella parte alta del riquadro servono per fissare i margini e gli stop per la tabulazione: il tutto viene attivato con un click su **Margins. Word Wrap**, invece, attiva la funzione



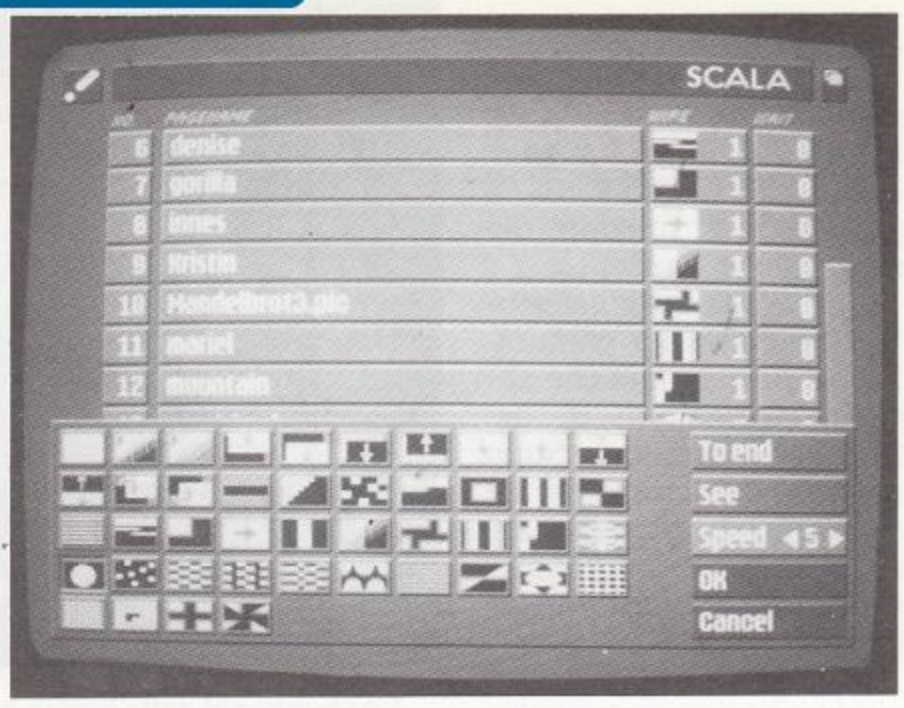
Il file requester di «Scala». Ognuno dei pulsanti in alto a destra è associato ad una particolare directory.

http://www.oldgamesitalia.net/
http://www.oldgamesitalia.net/

http://www.oldgamesitalia.net/



Il riquadro principale, che appare durante l'editing delle pagine e consente di accedere a numerosi sotto-menu.



Le 44 alternative per l'ingresso di un'immagine. Come si può notare, alcune sono davvero uniche.

omonima, riportando a capo l'intera parola se questa non rientra nel margine.

Save Layout memorizza la configurazione su disco per utilizzi futuri, mentre **Save Default** sostituisce il layout attuale a quello standard, quello cioè che viene utilizzato in mancanza di istruzioni specifiche. Ma torniamo, con **OK** o con **Cancel**, al riquadro di editing, che ci riserva ancora moltissime sorprese.

Il pulsante denominato **Load** accede ad un sotto-menu per mezzo del quale si può caricare un layout (ogni oggetto presente nella pagina può averne uno differente), un altro sfondo (senza perdere gli oggetti ad esso sovrapposti), un oggetto di tipo grafico, o **Symbol** (salvato con «Deluxe Paint») o con altro in formato IFF Brush, un testo (in formato ASCII). I Symbol sono anch'essi dotati di un proprio layout, e vengono considerati da «Scala» alla stessa stregua delle linee di testo. Tornando al riquadro principale, il pulsante con la scritta **Font** ci consente di accedere al riquadro di selezione del tipo di carattere.

Il programma è dotato di 14 font piuttosto belli (che nella versione beta in nostro possesso necessitano di alcuni ritocchi: mancano per esempio le lettere accentate ed alcuni simboli, e c'è qualche pixel fuori posto qua e là); a differenza del concorrente «Pro-Video», però, «Scala» utilizza il formato standard, per cui i font a disposizione sono virtualmente illimitati. Come prevedibile, il gadget **Net** attiva la griglia (le cui dimensioni sono state specificate nel Layout Editor); **Palette** serve per modificare la tavolozza dei colori, mentre **Move** ci consente di posizionare gli oggetti già presenti sullo schermo dove preferiamo.

LA STANZA DEI BOTTONI

Selezionando **Buttons** possiamo conferire alla nostra presentazione un minimo di interattività: tramite il riquadro che apparirà definiremo una serie di «hit box», ovvero delle zone rettangolari di schermo che agiranno come pulsanti. A ciascun pulsante è associata una pagina: se l'utente farà un click con il mouse puntato su di una hit box, la presentazione continuerà dalla pagina ad essa associata. Si può scegliere se la hit box debba passare o meno al negativo quando viene selezionata, ma purtroppo non si può attribuire al pulsante un'immagine alternativa.

Appare quindi evidente come «Scala» abbia una

LA CONFIGURAZIONE DI SCALA

Intervenendo con un buon editor ASCII sul file S:Scala.config si possono modificare alcuni parametri non inclusi nel menu System di «Scala». Di seguito riportiamo un esempio di configurazione: sono commentate soltanto le voci non modificabili da programma.

C1.0 Questa riga non deve essere toccata
USERLEVEL full Non ancora implementato
SCREENPALETTE 656 FED 000 989 DB0 47A 148 025
La palette dei menu (in formato RGB), per chi non si accontentasse delle 22 predefinite.

MODE page
CREATEICONS on
Gli script salvati da Scala devono avere un'icona?
DRIVE «Scala» «Network:dh0/2D-Graphics/Scala» 1
DRIVE «Pictures» «Network:Work/Pictures» 1

Questi sono i pulsanti definiti nel file requester: il primo parametro è il nome, il secondo il path associato, il terzo il colore.
WORKDIR ""

La directory di default su cui si aprirà il file requester
ERRORFILE «RAD:Error»

In caso di errori o problemi di funzionamento, informazioni utili per il debug saranno memorizzate in questo file

BUFFERMODE Dynamic
PAGECOUNTER 2
COUNTERDWELL 10
LACEALL off

In occasione di registrazioni su nastro, è bene attivare l'interlace anche su immagini in bassa e media risoluzione

POINTER off
MOUSE on
KEYBOARD on
JOYSTICK off

http://www.oldgamesitalia.net/
http://www.oldgamesitalia.net/

http://www.oldgamesitalia.net/

marcia in più rispetto a «Pro-Video», potendo essere utilizzato anche per pubblicità diretta (vetrine e cataloghi interattivi), oltre che per riversamenti su nastro.

Il pulsante **List** visualizza uno storyboard della pagina che si sta creando, come quello visibile in figura. Questa volta ogni riga rappresenta un oggetto o una linea di testo. Da sinistra verso destra, troviamo l'ordine in cui gli oggetti appariranno a video, il nome (o il testo) dell'oggetto, il tipo e la velocità di ingresso, i secondi di pausa.

Agendo con il mouse sulla colonna dei nomi, possiamo modificare l'ordine di apparizione: spostando una determinata riga in fondo alla lista, la riga apparirà per ultima, anche se la sua posizione fisica sullo schermo non verrà modificata.

GLI EFFETTI SPECIALI

Con un click su di una casella della terza colonna accederemo al menu delle transizioni (**Wipes**), visibile in figura: si può accedere a questo menu anche dal pannello principale, con il piccolo pulsante immediatamente alla sinistra di **List**.

I 41 pulsanti nella parte bassa selezionano ciascuno un tipo di transizione, eccetto il primo: con questo (selezionato di default) faremo sì che l'oggetto in questione compaia sullo schermo fin dall'inizio.

Per tutte le altre transizioni, con **Speed** si indica la velocità di esecuzione: da 1 (molto lento) a 10 (fulmineo).

Se prima di uscire dal sottomenu (con **OK**) si selezionerà il pulsante con la scritta **To End**, la transizione indicata sarà utilizzata non solo per l'oggetto attualmente selezionato, ma anche per tutti i seguenti, fino alla fine dello storyboard.

PAUSE E ANTEPRIME

L'ultima colonna, intitolata **Wait**, non dovrebbe presentare alcun problema: è sufficiente indicare un intervallo di pausa in secondi. Anche qui si può attivare l'utilissima funzione **To End**.

Il pulsante **See**, presente in moltissimi menu e sottomenu, ha sempre la funzione di «anteprima» dell'animazione della pagina, con le ultime modifiche da noi apportate.

Tornati al pannello principale, ne usciamo con **Finished**. Dopo aver assegnato un nome alla pagina (o accettato quello che «Scala» ci propone), abbiamo due possibilità: tornare allo storyboard principale (con **OK**) o creare un'altra pagina con il medesimo sfondo (con **Save and New**).

Scegliamo la prima opzione, e finalmente il nostro storyboard principale non sarà più vuoto.

Sono ora attivate tutte le funzioni che abbiamo visto nello story, più una: con un click sui numeri della prima colonna potremo copiare una pagina in un'altra posizione.

44 ORIGINALI TRANSIZIONI

La colonna **Wipe** indica stavolta la transizione di ingresso della pagina di sfondo. Questa può essere scelta tra 44 possibilità, alcune delle quali davvero originali: basta dare un'occhiata alle immagini per rendersene conto.

Le transizioni sono tutte bidimensionali, ma realizzate con tanta cura e fantasia da non sfigurare in produzioni video di qualità commerciale.

Il significato dei pulsanti nella parte bassa dello schermo è abbastanza intuitivo: **See** esegue una parte dello slide show, **Change** consente di modificare una pagina già esistente, **Delete** elimina una o più pagine.

Load Script e **Save Script** curano il trasferimento dei dati da e verso le memorie di massa. Una caratteristica molto interessante è che gli script vengono salvati in formato Ascii puro, cosicché è addirittura possibile modificarli senza l'utilizzo del programma principale: basta un buon editor, oltre che la conoscenza del linguaggio utilizzato.

Una presentazione di questo genere si realizza in pochi minuti. I disegni negli angoli sono Symbols.

Per mezzo del Layout Editor, ad ogni oggetto sullo schermo vengono associati parametri propri.

ATTENZIONE!
I contenuti della presente videocassetta sono protetti da copyright.
Ne sono vietate la duplicazione e la diffusione anche parziali
copyright 1991 L. Mirabelli

Un'animazione simile a questa conferirà un tono professionale alle nostre videocassette.

http://www.oldgamesitalia.net/
http://www.oldgamesitalia.net/

http://www.oldgamesitalia.net/

QUALCHE CONSIGLIO PRATICO

Davanti ad un programma così potente, è facile cedere alla tentazione di utilizzare più transizioni e più font possibile. Il risultato è che spesso chi vede una presentazione è disorientato, e ha l'impressione di assistere ad un collage realizzato con pezzi reperiti chissà dove. Per chi si accosta per la prima volta a questo genere di lavoro, dunque, la prima raccomandazione è: semplicità. Non usate più di due font (uno per i titoli ed uno per tutto il resto), e più di due «famiglie» di transizioni (ad esempio gli effetti mosaico e gli spostamenti diagonali): a conferire varietà alla presentazione basteranno caratteri di corpo differente o qualche transizione più veloce delle altre, oltre (ci auguriamo) che i contenuti del vostro messaggio.

Tutte le attività commerciali possono trarre profitto da «Scala» (a patto di cambiare il menu...).

Per la verifica delle temporizzazioni, fate partire la presentazione e leggete ad alta voce tutto ciò che appare sullo schermo: se qualche scritta scompare prima che abbiate finito di leggerla, è consigliabile prolungarne il tempo di permanenza. Ricordate che un messaggio letto molto rapidamente viene memorizzato con molta difficoltà.
Gli sfondi di «Scala» sono meravigliosi, ma se prevedete di realizzare più di un lavoro procuratevene o realizzatene altri, o almeno modificatene i colori.

Se dovete sovrapporre delle scritte ad un'immagine a colori, passatela al bianco e nero o createne una versione Line Art (avente cioè i soli contorni) con un programma tipo «Pixmate» o «The Art Department»: eviterete così che i caratteri scompaiono, fagocitati da uno sfondo più evidente di loro.
A proposito di colori: pochi e ben definiti! Se però dovete registrare su nastro, ricordate che più i colori sono vivaci, più sono indefiniti i contorni dell'oggetto. Da evitare come la peste il rosso pieno: poiché viene registrato con la frequenza più elevata, le sbavature sono terribili (e non solo su VHS).

UN PROGRAMMA SU MISURA

Il pulsante **System** accede al pannello di controllo del sistema, per mezzo del quale ogni utente può personalizzare il programma a seconda delle sue esigenze e della sua configurazione.

Con il primo pulsante, denominato **Pointer**, potremo scegliere se il puntatore del mouse debba essere presente o meno sullo schermo durante la presentazione. Con **Counter** sceglieremo se e dove debba comparire il contatore del numero di pagina, e con **Counter Dwell** determineremo i secondi di permanenza dello stesso.

Menu Colors permette di personalizzare i colori dell'interfaccia utente. Attenzione, però: questo non avviene mediante un riquadro Palette, come in tanti altri programmi, bensì scegliendo da un campionario di addirittura 22 set di colori già pronti. Questa sì che è classe!

Pagebuffer è un'area di memoria nella quale viene pre-caricata l'immagine successiva. Con l'omonimo pulsante quest'ultima può essere resa statica (cioè allocata una volta per tutte, se c'è Ram a sufficienza) dinamica (allocata pagina per pagina solo se la memoria lo consente), o disattivata.

I pulsanti **Pages, Lines** e **Loops** sono le tre alternative che «Scala» ci mette a disposizione per la nostra presentazione. Scegliendo la prima, alla fine di ogni pagina il programma attenderà un click del mouse per proseguire; con la seconda, occorrerà un click dopo ogni linea di testo od oggetto; con la terza, invece, la presentazione procederà automaticamente fino alla pressione del tasto <Esc> o al raggiungimento del numero di cicli specificato.

Interactive seleziona il mezzo di input durante la presentazione: mouse, joystick, o nessuno.

Il pulsante **Close Workbench** racimola un po' di preziosa Chip RAM; con **System Information**, nel riquadro con i nomi dei programmatori compariranno informazioni sullo stato della RAM e su quello delle memorie di massa.

Save Configuration memorizza il tutto in un file chiamato Scala.config, nella directory S:

Anche qui si tratta di un file ASCII, editando il quale si scoprono altri parametri che è possibile modificare (vedi riquadro).

È evidente che il giudizio su «Scala» è più che positivo: siamo di fronte ad un software che unisce la perfezione tecnica (le transizioni sono tutte velocissime e realizzate in double buffering in modo impeccabile) a mille attenzioni per il suo utente, al punto che utilizzarlo è un vero piacere. □

Un Amiga dotato di «Scala» e Genlock può sostituire quasi completamente una titolatrice video.

http://www.oldgamesitalia.net/
http://www.oldgamesitalia.net/

http://www.oldgamesitalia.net/

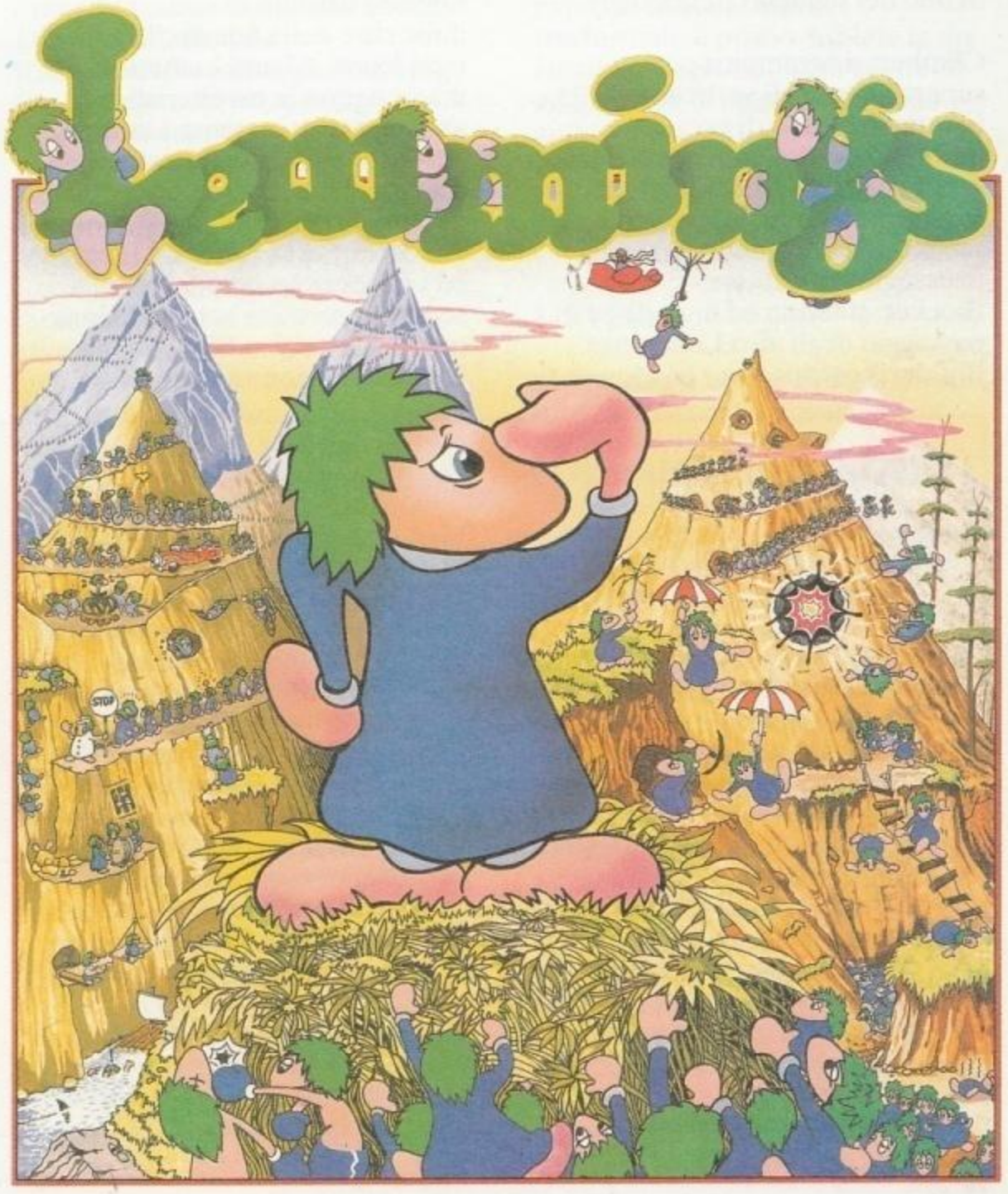
a cura di Vertigo

LEMMINGS

I Lemmings, protagonisti dell'omonimo simpaticissimo gioco della Psygnosys, sono minuscoli omiciattoli azzurri con i capelli verdi che, per qualche imperscrutabile motivo, sono irresistibilmente votati al suicidio.
Per realizzare questa insana ambizione, essi si precipitano in massa in luoghi costellati di trappole e pericoli, sperando di poter porre fine ai propri giorni gettandosi in qualche voragine o facendosi ridurre a carne trita da una dei tanti marchingegni disseminati per tutti i livelli.
Sta all'abilità del giocatore riuscire ad evitare l'ecatombe ed a trarre in salvo tutti gli stupidi animali, guidandoli verso l'apertura presente in ognuno dei 160 livelli di gioco.
Ogni Lemming entra in un livello attraverso l'apposita botola, ed è inizialmente definito come appartente alla categoria degli Walker: comincia a camminare senza interruzione, incurante dei pericoli e, quando incontra un ostacolo insormontabile, si limita a cambiare direzione senza fermarsi.
Se i Lemmings s'imbattono in un baratro, in una trappola o in un altro pericolo letale, ci si precipitano senza esitazione, suicidandosi in massa. Lo scopo del gioco consiste nel riuscire a salvare il maggior numero possibile di Lemmings entro il tempo limite assegnato, e di condurli fino all'uscita facendo loro superare tutti gli ostacoli.

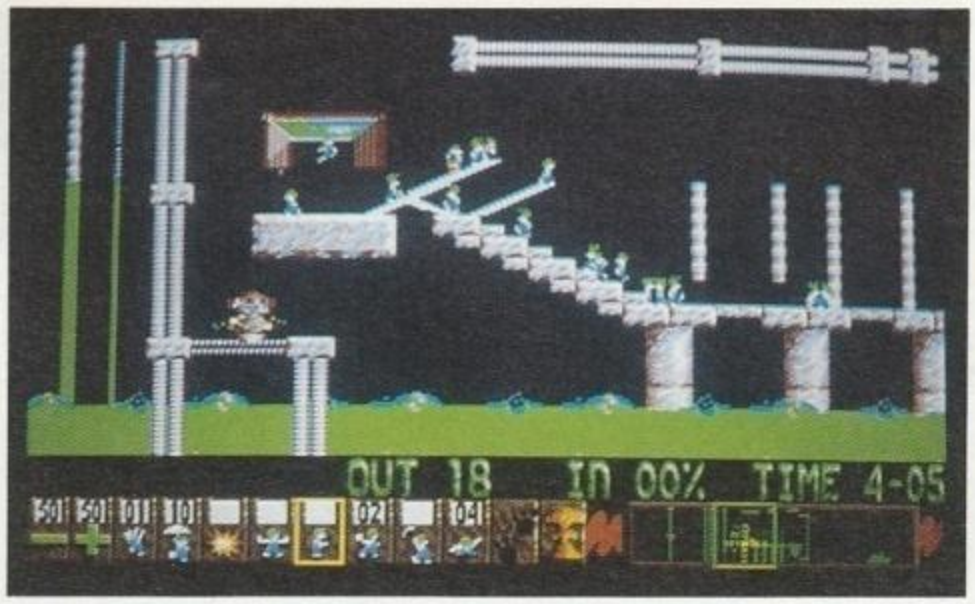
LE ICONE

Il giocatore dispone di una serie di icone, selezionabili con il mouse, che consentono di dotare ogni Lemming di una o più abilità particolari, quali la capacità di scavare, quella di costruire ponti o quella di arrampicarsi lungo pareti impervie.
Ogni livello richiede strategia e prontezza di riflessi, per pianificare le azioni di ogni Lemming e riuscire a dirigere i movimenti del branco, evitandone il suicidio di massa.
Sfortunatamente, le cose sono complicate dalla velocità alla quale i Lemmings si precipitano in ogni livello, e dalla difficoltà di dover spesso agire contemporaneamente in più punti distanti, per coordinare le azioni dei malefici omuncoli.
La parte superiore dello schermo è occupata dal territorio di gioco vero e proprio, che può scrollare orizzontalmente tramite il mouse.
La parte inferiore del video comprende invece sulla destra una sorta di mini radar, che mostra schematicamente il contenuto del livello e la posizione dei Lemmings, sotto forma di puntini luminosi. Sulla sinistra sono situate invece dodici icone, corrispondenti ad alcuni comandi ed alle caratteristiche che possono essere assegnate ad ogni Lemming.
Le prime due icone,



http://www.oldgamesitalia.net/
http://www.oldgamesitalia.net/

http://www.oldgamesitalia.net/



contrassegnate dai simboli «+» e «—», consentono di variare la velocità alla quale i Lemmings si riversano dalla botola. La velocità iniziale stabilita all'inizio di ogni livello è fissa e non può essere diminuita, ma solo aumentata.
Le successive otto icone rappresentano graficamente altrettanti tipi di Lemming. Selezionando un'icona e successivamente clickando sulla «vittima» prescelta, essa si trasformerà da semplice Walker in uno dei seguenti Lemming:
Climber: si arrampica per superare ostacoli verticali;
Floater: è dotato di un ombrello-paracadute per cadere senza danni da grandi altezze;
Bomber: esplode dopo cinque secondi, distruggendo parte del paesaggio circostante;
Blocker: si ferma ed impedisce il passaggio degli altri Lemmings;
Builder: costruisce un ponte diagonalmente, per superare baratri ed ostacoli;
Basher: scava orizzontalmente attraverso gli ostacoli;
Miner: scava diagonalmente verso il basso con un piccone;
Digger: scava verticalmente verso il basso.

LE CARATTERISTICHE

Ognuna di queste caratteristiche può essere applicata per un numero limitato di volte, indicato dalle cifre nella finestrella sopra ogni icona. Alcuni Lemmings mantengono le caratteristiche assegnate per un tempo limitato, altri le conservano fino allo scadere del tempo o fino a quando non muoiono.
Ad esempio, i Climber, i Floater ed i Blocker restano tali per sempre una volta assegnate loro queste capacità; i Blocker, in particolare, continuano ad ostacolare il passaggio degli altri Lemmings fino a quando non vengono eliminati trasformandoli in Bomber e facendoli quindi esplodere.
I Digger, i Basher ed i Miner continuano invece a scavare soltanto finché non incontrano un ostacolo resistente o il vuoto; la loro opera di scavo può però essere interrotta trasformandoli in Blocker o assegnando loro qualche altra capacità.
I Builder sono dotati di un numero limitato di mattoni da costruzione (dodici), terminati i quali cessano di costruire e ricominciano a camminare. Un Builder smette inoltre di costruire quando lui o il suo ponte urtano un ostacolo.
Un Climber può diventare Floater senza perdere la prima caratteristica, e viceversa. il Lemming risultante dall'unione di queste due capacità si chiama Athlete ed è molto potente, perché può superare quasi ogni ostacolo (anche molto alto) scalandolo e lasciandosi cadere con il paracadute.

I CODICI DI LEMMINGS

Livelli per un giocatore

FUN	TRICKY
1 =	1 = HBANLMFPDV
2 = IJHLDJBCCW	2 = BINLMFJQDQ
3 = NHLDHBADCR	3 = BAJHLDIBEO
4 = HLDHBINECK	4 = IJHLDIBCEX
5 = LDHBAJLFCT	5 = NHLDIBADEU
6 = DHBIJLLGCM	6 = HLDIBINEEN
7 = HBANLLDHCJ	7 = LDIBAJLFEW
8 = BIMLLDHICS	8 = DIBIJLLGEP
9 = BAJHMLHJCM	9 = IBANLLDHEM
10 = IJHMDHBKCN	10 = BINLLDIIEV
11 = NHMDHBALCK	11 = BAJHMDIJEX
12 = HMDHBINMCT	12 = IJHMDIBKEQ
13 = MDHBAJLNCM	13 = NHMDIBALEN
14 = DHBIJLMOCV	14 = HMDIBINMEW
15 = HBANLMDPCS	15 = MDIBAJLNEP
16 = BINLMDHQCL	16 = DIBIJLMOEY
17 = BAJHLFHBOO	17 = IBANLMDPEV
18 = IJHLFHBCDX	18 = BINLMDIQEO
19 = NHLFHBADDU	19 = BAJHLFIBFR
20 = HLFHBINEDN	20 = IJHLFIBCFK
21 = HLFHBJLFDW	21 = NHLFIBADFX
22 = FHBIJLLGDP	22 = HLFIBINEFQ
23 = HBANLLFHDM	23 = LFIBAJLFFJ
24 = BINLLFHIDV	24 = FIBIJLLGFS
25 = BAJHMFHJDX	25 = IBANLLFHFP
26 = IJHMFHBKDQ	26 = BINLLFIIFY
27 = NHMFHBALDN	27 = BAJHMFIJKF
28 = HMFHBINMDW	28 = IJHMFIBKFT
29 = MFHBAJLNDP	29 = NHMFIBALFQ
30 = FHBIJLMODY	30 = HMFIBINMFJ

TAXING	MAYHEM
1 = MFIBAJLNFS	1 = NHMFHFAKHV
2 = FIBIJLMOFL	2 = HMFHFINMHO
3 = IBANLMFPFY	3 = MFHFAJLNHX
4 = BINLMFIQFR	4 = FHFIJLMOHQ
5 = FAJHLDHBGT	5 = HFANLMFPHN
6 = IJHLDHFCGM	6 = FINLMFHQHW
7 = NHLDHFADGJ	7 = FAJHLDIBIW
8 = HLDHFINEGS	8 = IJHCDIFCIP
9 = LDHFAJLFGL	9 = NHLDIFADIM
10 = DHFIJLLGGU	10 = HLDIFINEIV
11 = HFANLLDHGR	11 = LDIFAJLFIO
12 = FINLLDHIGK	12 = DIFIJLLGIX
13 = FAJHMDHJGM	13 = IFANLLDHIV
14 = IJHMDHFKGV	14 = FINLLDIIIN
15 = NHMDHFALGS	15 = FAJHMDIJIP
16 = HMDHFINMGL	16 = IJHMDIFKIY
17 = MDHFAJLNGU	17 = NHMDIFALIV
18 = DHFIJLMOGN	18 = HMDIFINMIO
19 = HFANLMDPGK	19 = MDIFAJLNIX
20 = FINLMDHQGT	20 = DIFIJLMOIQ
21 = FAJHLFHBHW	21 = IFANLMDPIN
22 = IJHLFHFCHP	22 = FINLMDIQIW

http://www.oldgamesitalia.net/
http://www.oldgamesitalia.net/

http://www.oldgamesitalia.net/

Le due icone finali raffigurano rispettivamente le impronte di un paio di zampe ed il fungo di un'esplosione nucleare: alla prima corrisponde una funzione di pausa, che congela il movimento dei Lemmings, consentendovi però di spostare il mouse lungo lo schermo per esaminare il livello. La seconda icona, se clickata rapidamente due volte, fa esplodere in maniera spettacolare tutti i Lemmings e viene usata in situazioni disperate per ricominciare il livello dall'inizio. La linea di testo sopra le icone, infine, mostra il nome ed il numero di Lemmings attualmente situati sotto il cursore, il numero di quelli presenti nel livello in quel momento, la percentuale di Lemmings tratti in salvo ed il tempo rimanente. All'inizio di ogni livello viene visualizzato il numero totale di Lemmings e la percentuale minima di essi che deve essere salvata per poter passare al livello successivo.

UNA STRATEGIA

È difficile elaborare una strategia valida in tutte le circostanze, vista la quantità di problemi da risolvere e la varietà di ostacoli diversi da superare.
Può essere utilissimo ricorrere spesso alla funzione di pausa (attivabile anche con il tasto P) per spostarsi rapidamente da una parte all'altra di ogni livello senza perdere tempo. Congelare il movimento dei Lemmings consente di far scorrere lo schermo comodamente senza doversi preoccupare della sorte dei mostriciattoli situati alle estremità del piano di gioco, e permette di risparmiare molti secondi preziosi.
Spesso, poi, la soluzione apparentemente più evidente non è quella corretta: in molti casi certi ostacoli complicati possono essere superati passandoci sotto, facendo scavare verticalmente un Digger nel terreno e tramutandolo, prima che buchi il fondo dello schermo, in un Basher. In questo modo potrete scavare comodamente un tunnel sotterraneo.
Fate attenzione anche a dove interrompete il flusso dei Lemmings con i Blocker: dal momento che l'unico metodo per eliminarli consiste nel farli esplodere, occorre pensare bene a dove li si fa fermare. Se mettete un Blocker su di una passerella o su di una striscia di terreno molto sottile, l'esplosione causerà una voragine attraverso la quale precipiteranno tutti i Lemmings che premevano alle sue spalle.
Ricordate che all'occorrenza potete interrompere l'azione di un Lemming scavatore, per evitare che precipiti nel vuoto, trasformandolo in qualcos'altro (ad es. in un Builder): gli scavatori, inoltre, possono passare soltanto attraverso il terreno o le mura, ma non possono scavare nel metallo.
Spesso uno dei problemi è costituito dalla velocità alla quale i Lemmings si riversano in un livello. Un metodo per rallentarne la marcia in mancanza di Blocker può essere quello di trasformarli in Builder e far costruir loro brevi tratti di ponte apparentemente inutili: questa operazione rende più lenti i movimenti dei Lemmings e ne frena l'impeto per qualche istante, spesso vitale.
Le mura o gli ostacoli contrassegnati da frecce possono essere scavati solo nella direzione segnalata, che naturalmente è quasi sempre opposta a quella dalla quale i Lemmings arrivano: in questi casi servono di solito due Lemmings, da trasformare in Athlete per far loro scalare e superare l'ostacolo. Una volta giunti al di là del muro, trasformate il primo Athlete in un Blocker: il secondo sarà costretto ad invertire la marcia e, appena giunto a contatto del muro, potrà essere mutato in un Basher per scavare un provvidenziale tunnel.

I LIVELLI

«Lemmings» comprende 20 livelli da giocare contemporaneamente in due, e 140 per un solo giocatore. Questi ultimi livelli sono suddivisi in quattro gruppi di trenta livelli ciascuno, in base alla difficoltà: Fun, Tricky,.Taxing e Mayhem. I livelli Fun sono destinati ai principianti, e consentono al giocatore di fare pratica con i diversi tipi di Lemming e con le loro caratteristiche; i restanti livelli costituiscono il gioco vero e proprio, e richiedono spesso impegno ed abilità quasi sovraumane per essere portati a termine.
Al termine di ogni livello viene mostrata una password di dieci caratteri, da usare per poter riprendere una partita a distanza di tempo senza dover iniziare ogni volta da capo.

□

23 = NHLFHFADHM	23 = FAJHLFIBJJ
24 = HLFHFINEHV	24 = IJHLFIFCJS
25 = LFHFAJLFHO	25 = NHLFIFADJP
26 = FHFIJLLGHX	26 = HLFIFINEJY
27 = HFANLLFHHU	27 = LFIFAJLFJR
28 = FINLLFHIHN	28 = FIFIJLLGJK
29 = FAJHMFHJHP	29 = IFANLLFHJX
30 = IJHMFHFKHY	30 = FINLLFIIJQ

Livelli per due giocatori

1 = JAJHLDIBMO	11 = NHMDIJALMN
2 = IJHLDKJCMJ	12 = HMDIJINMMW
3 = NHLDKJADMW	13 = MDIJAJLNMP
4 = HLDIJINEMN	14 = DIJIJLMOMY
5 = LDIJAJLFMW	15 = IJANLMOPMV
6 = DIJIJLLGMP	16 = JINLMDIQMO
7 = IJANLLDHMM	17 = JAJHLFIBNR
8 = JINLLDIIMV	18 = IJHLFIJCNK
9 = JAJHMDIJMX	19 = NXLFIJADNX
10 = IJHMDIJKMQ	20 = HLFIJINENQ

http://www.oldgamesitalia.net/
http://www.oldgamesitalia.net/

http://www.oldgamesitalia.net/

Imagine: ray tracing a 24 bit

L'erede di «Turbo Silver» offre prestazioni mai viste in precedenza tra i programmi di grafica ed animazione in ray tracing, con una marcia in più: sedici milioni di colori.

di GUIDO QUARONI

Dopo diversi mesi di intenso lavoro la **Impulse**, nota software house americana, ha recentemente immesso sul mercato «**Imagine**», un potente pacchetto di grafica **ray tracing** per Amiga che inaugura una nuova generazione di programmi di modellazione ed animazione tridimensionale.

Nato come successore del popolarissimo «**Turbo Silver**», con il quale mantiene un'ottima compatibilità verso il basso per quanto riguarda il formato dei file, «Imagine» è infatti tra i pochi programmi di grafica tridimensionale in grado di gestire un numero di colori superiore a quelli normalmente visualizzabili da Amiga, fino ad un massimo di oltre 16 milioni di sfumature.

«Imagine» è, come di consueto per questo genere di software, disponibile in due versioni: una per i modelli Amiga dotati del normale processore 68000 e l'altra per i più potenti computer provvisti di 68020/30 con coprocessore matematico (come ad esempio Amiga 3000).

La filosofia del programma è radicalmente mutata rispetto ai precedenti concorrenti, compreso «Turbo Silver»: la creazione degli oggetti avviene mediante due editor specifici (**Detail** e **Forms Editor**); un terzo ambiente di lavoro permette la creazione e la gestione dei **Cycle Objects** (oggetti animati, simili agli AnimBrush dei programmi di grafica bidimensionale); la stesura dello storyboard di una sequenza animata viene gestita da un ulteriore editor

L'effetto «acqua» di questa immagine è reso possibile grazie al programma «Waves 2.1», che genera superfici di questo tipo.

http://www.oldgamesitalia.net/

http://www.oldgamesitalia.net/

http://www.oldgamesitalia.net/

(**Stage Editor**); il rendering vero e proprio viene eseguito nel **Project Editor**.

LA FILOSOFIA DI IMAGINE

A prima vista, questa suddivisione degli ambienti di lavoro può sembrare poco consistente, ma basta qualche ora di pratica per capire ed apprezzare le potenzialità offerte da questa struttura del programma.

«Imagine» opera, utilizzando i noti algoritmi di ray tracing, a 12 o 24 bitplane (4096 o 16.7 milioni di tinte contemporanee) con una risoluzione variabile fino ad un massimo di 8000 per 8000 punti (risoluzione adatta al trasferimento su pellicola cinematografica). Sono ovviamente supportate tutte le risoluzioni standard di Amiga e l'uscita diretta sulla nuova scheda video PAL «**FireCracker 24**» a 24 bitplane per Amiga 2000, prodotta dalla stessa Impulse.

I fotogrammi possono venir memorizzati nei formati IFF standard, RGB8, o RAW RGB, ed eventualmente compattati secondo lo standard IFF ANIM-5 o in un formato particolare proprio di «Imagine», denominato RGBN.

Il rendering può essere effettuato a differenti livelli di precisione ed accuratezza. Si possono ottenere immagini in modo **Wire Frame** (con i soli contorni visibili), **Solid Model** (con contorni e lati pieni) o in **Ray Tracing** (generando tutte le ombre e le sfumature di colore). I tempi di calcolo sono proporzionali alla risoluzione ed alla tecnica di rendering cui si ricorre: il modo Wire Frame, essendo rapidissimo, viene utilizzato in genere per visualizzare schematicamente un'immagine che potrà poi essere generata più dettagliatamente ricorrendo al Ray Tracing vero e proprio. «Imagine» è comunque molto rapido, ed i tempi del *rendering* sono sempre più brevi rispetto ai programmi concorrenti. Dopo il caricamento, «Imagine» visualizza il Project Editor, dove, per poter iniziare una sessione di lavoro, è necessario creare un nuovo progetto od eventualmente aprirne uno già presente su disco; il programma memorizzerà tutte le informazioni necessarie per le descrizioni delle scene e degli eventuali sottoprogetti in una specifica directory nella quale si potranno impostare tutti i parametri di visualizzazione visti in precedenza (la risoluzione, il tipo di rendering, etc.). A questo punto sarà possibile accedere ai restanti quattro editor già menzionati.

Un'immagine in modo Color Flat Shading. Per un veloce preview si può optare per un rendering meno realistico, ma ugualmente efficace.

DETAIL EDITOR

L'editor di oggetti di «Imagine» è senza dubbio il più potente tra quelli presenti oggi per Amiga. L'editing ha luogo tramite un metodo di rappresentazione visiva che viene definito **Quadri-View**: le consuete tre viste ortogonali vengono affiancate da una quarta vista prospettica per facilitare la visualizzazione dei solidi. È comunque possibile lavorare sulle quattro viste indipendentemente (nella finestra 3D si può rappresentare l'oggetto utilizzando un semplice algoritmo di **Flat Shading** a tonalità di grigio). L'approccio è per certi versi simile a quello già visto in «Turbo Silver», con il quale «Imagine» presenta ovviamente molti punti di contatto.

Ogni oggetto è costituito da un asse, che ne determina orientamento e posizione, e da un insieme di triangoli interconnessi allo scopo di formare superfici complesse. Sono presenti inoltre due tipi particolari di oggetti, la **sfera** geometrica ed il **ground** (terreno), già visti in «Turbo Silver». È inoltre disponibile una serie di primitive grafiche quali il cono, il cilindro, il toroide e così via, che possono essere generate automaticamente dal programma.

Nel **menu Mode** possiamo decidere se lavorare sugli oggetti, sulle superficie o sui singoli punti che compongono un solido. La selezione di punti, lati o facce viene effettuata con il classico click del mouse o utilizzando il *drag box* o una linea tracciata a mano libera per selezioni multiple. Lo spostamento di uno o più vertici viene effettuato in tempo reale con una notevole fluidità, così come la rotazione e la traslazione di un intero oggetto.

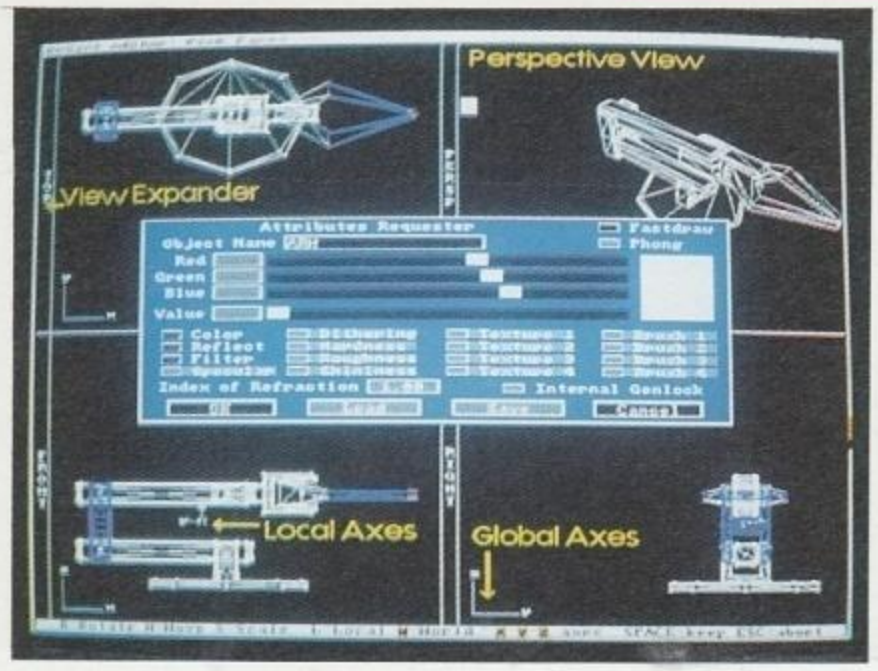
Durante queste opera-

Nel Project Editor vengono definiti tutti i parametri relativi al rendering.

http://www.oldgamesitalia.net/

http://www.oldgamesitalia.net/

http://www.oldgamesitalia.net/



Detail Editor: dopo l'editing è possibile intervenire sulle caratteristiche della superficie di uno o più poligoni.

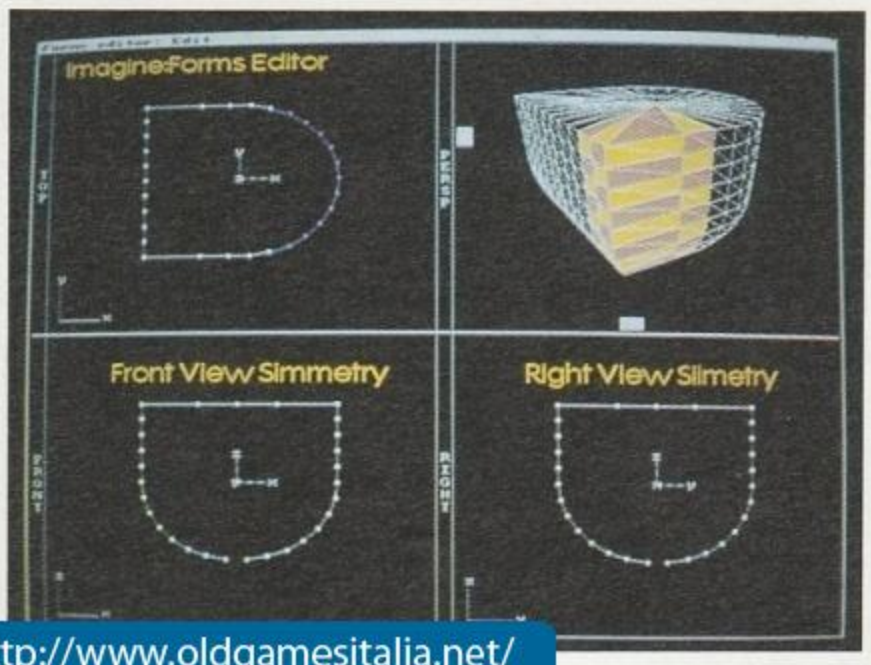
zioni il solido non viene mostrato nei dettagli, bensì viene rappresentato schematicamente da un parallelepipedo (definito **bounding box**) per rendere più veloci tutte le operazioni. Grazie a questo stratagemma, possiamo tranquillamente dimenticare le lunghe attese per il redraw completo di tutta la scena anche dopo il semplice spostamento di un singolo punto caratteristiche di molti programmi di grafica 3D.

Le rotazioni, anch'esse gestite in tempo reale, possono essere locali (ovvero riferite all'asse dell'oggetto selezionato) o globali (riferite al sistema di coordinate assolute); inoltre, se necessario, possono influire solo sulla posizione dell'asse, determinando così l'orientamento del corpo rispetto al sistema assoluto dell'editor.

Sono accessibili, tramite il **menu Mold**, potenti tool per l'editing di oggetti complessi. Si può compiere l'estrusione di una linea piana chiusa lungo una retta (verticale od obliqua) o lungo una curva (insieme di segmenti interconnessi) precedentemente creata; durante l'estrusione possiamo inoltre ruotare la superficie, traslarla e scalarla rispetto alla linea «guida».

Attraverso i comandi **Spin** e **Sweep** è possibile creare superfici di rotazione (chiuse o aperte nel centro) partendo da un profilo di una o più linee. Infine, sono presenti comandi che permettono il *wrapping* (avvolgimento) totale o parziale di un oggetto intorno ad una sfera o ad un cilindro.

Forms Editor: con questo editor si possono ottenere solidi di rotazione complessi con estrema facilità.



«Imagine» è in grado di importare un'immagine IFF, possibilmente ad un solo bitplane, e di trasformarla in un oggetto bidimensionale mantenendo un livello di precisione accettabile. Con i comandi **Group** ed **UnGroup** è possibile «raggruppare» una serie di oggetti in un'unica entità gerarchica, per meglio gestirli.

SEZIONI E TRASFORMAZIONI

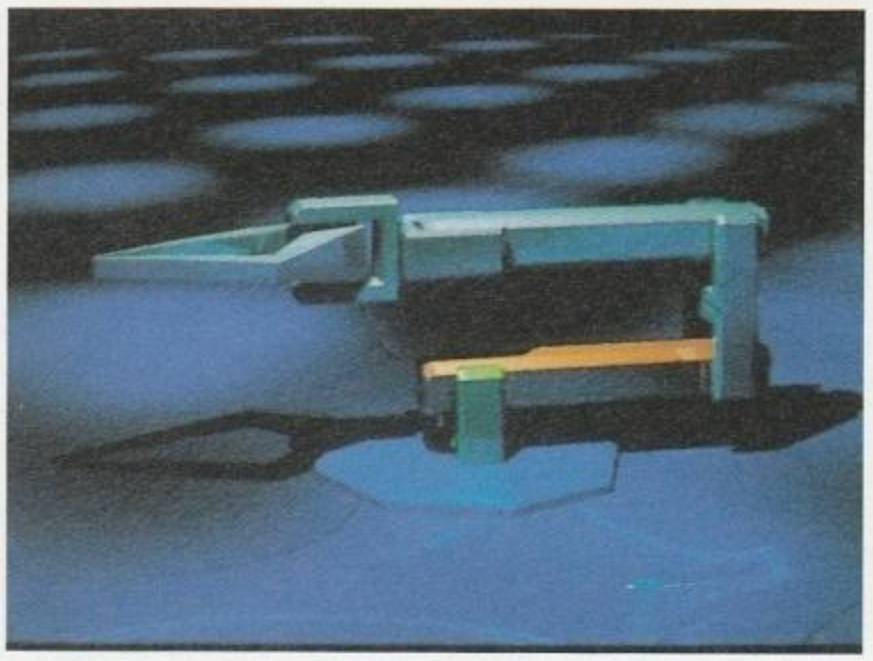
Sono disponibili i classici comandi «taglia ed incolla» ed il potentissimo ed inedito **Slice**, che permette di effettuare operazioni di tipo booleano tra due oggetti. Selezionando contemporaneamente due solidi (si utilizza il tasto Shift in combinazione con il mouse) aventi punti di contatto e di intersezione ed attivando la funzione Slice, il programma ricaverà una serie di oggetti, aventi asse proprio, ottenuti per somma e sottrazione booleana tra le parti a contatto.

Risulta quindi possibile tagliare una sfera con un piano o con qualsiasi altro solido, ed ottenere sezioni difficilmente realizzabili con l'editor tradizionale. Per questo tipo di operazioni logiche, «Imagine» richiede comunque una considerevole quantità di memoria.

Le trasformazioni possono essere fatte direttamente in una delle tre viste ortogonali con il mouse, o tramite uno specifico requester. Un tool senz'altro utile risulta inoltre essere il magnete, che rende possibile la creazione di forme molto particolari di linee stile Bezier.

La parte più interessante ed unica di questo programma è l'**Attribute Requester**, in cui vengono impostati tutti i parametri relativi alle caratteristiche superficiali di un oggetto. I poligoni possono essere resi con un livello di definizione mai visto prima su Amiga; si possono inoltre gestire fino a quattro **Texture** e quattro **Brush Wrap** contemporaneamente, tenendo conto del fatto che il *brush wrapping* (ovvero l'avvolgimento di un'immagine IFF intorno ad un oggetto) può essere effettuato in svariati modi.

Tra le opzioni del **Brush Wrap Requester** troviamo anche l'**Altitude Mapping**, che ci permette di ottenere il famoso effetto **Bump Mapping** (effetto bassorilievo), implementato per la prima volta su Amiga da «Imagine».



Un'immagine ray-traced. La texture assegnata al terreno è di tipo «disturbed» con proprietà parzialmente riflettenti.

http://www.oldgamesitalia.net/
http://www.oldgamesitalia.net/

http://www.oldgamesitalia.net/

IL FUTURO A 24 BIT

di Vincenzo Marangoni

Sotto il profilo delle prestazioni grafiche, un normale computer Amiga è «limitato» alle gestioni di immagini aventi una risoluzione massima di 640x512 punti ed un massimo di 4096 colori contemporaneamente (modo HAM o Dynamic Hi-Res).
Questi valori massimi, di tutto rispetto considerando il rapporto qualità-prezzo del computer, possono risultare insufficienti nel caso si voglia realizzare grafica computerizzata in campo professionale, per il riversamento su nastro magnetico o su pellicola fotografica. Per chi sente l'esigenza di migliorare le già notevoli prestazioni grafiche di Amiga, sono disponibili (o stanno per esserlo) numerose schede grafiche per Amiga 2000/3000 a 24 bit, che aumentano la quantità di colori visualizzabili contemporaneamente.
La maggior parte di questi prodotti, di origine americana, non risulta al momento già in distribuzione in Europa, essendo infatti necessaria la conversione per il supporto del formato PAL. Le prestazioni ed i prezzi indicativi riportati dovrebbero essere comunque sufficienti per fornire un'idea orientativa di cosa offre il mercato della grafica a 16 milioni di colori.

A2410: annunciata da Commodore, questa scheda a 24 bit per Amiga 2000/3000 deve ancora vedere ufficialmente la luce. Secondo le più recenti indiscrezioni dovrebbe comprendere un framebuffer con 2 Mb di Ram e fornire un segnale RGB in output.

FireCracker 24: prodotta dalla Impulse rappresenta, come si è detto, il completamento ideale ad «Imagine». Supporta i formati RGBN e IFF-24 ed è in grado di fornire un segnale video RGB a 24 bit di alta qualità per il collegamento con un monitor. Il costo è di circa 1500$ (con 2 Mb di memoria). La Impulse ha annunciato la prossima disponibilità di «Light», un software di grafica pittorica dedicato per il FireCracker.

Video Toaster: più orientato verso l'utilizzo in ambito DeskTop Video grazie al GenLock incorporato, questo prodotto della NewTek ha riscosso negli USA un successo strepitoso grazie al prezzo relativamente contenuto (circa 1500$) ed alle eccellenti prestazioni. Dotato di frame grabber, di un generatore di effetti video, di un potente software di raytracing a 24 bit («LightWave 3D») e di due frame buffer a 24 bit, il Video Toaster presenta l'unico inconveniente di non essere ancora disponibile in versione PAL per il mercato europeo, anche se l'attesa non sembra ormai destinata a prolungarsi ulteriormente. L'output RGB non è supportato.

Harlequin: addirittura 32, e non 24, sono i bitplane gestiti da questa scheda. In realtà le immagini sono sempre a 16 milioni di colori, in quanto gli 8 bit addizionali sono usati per la memorizzazione di informazioni per la gestione via software dei dati. La scheda, disponibile in varie versioni con frame buffer da un minimo di 1,5 mega ad un massimo di 4 mega di memoria, è prodotta in Scozia dalla ACS e costa intorno alle 2000 sterline (per il modello con 2 mega). L'output video è RGB, e sono supportati sia il formato video PAL che NTSC.

DCTV: la particolarità di questa scheda a 24 bit, prodotta dall'americana Digital Creation, consiste nello sfruttamento della memoria interna di Amiga, invece che di un Frame buffer separato, per la gestione delle immagini. La scheda supporta il formato IFF-24 ed è in grado di digitalizzare immagini provenienti da una videocamera o da un videoregistratore. Non essendo dotata di frame grabber, la scheda richiede una decina di secondi per la digitalizzazione e può quindi digitalizzare solo immagini statiche. Il segnale video in uscita è di tipo composito, ma purtroppo solo in formato NTSC. Il prezzo, negli USA, è di circa 500$.

HAM-E: a chi non può permettersi una scheda vera e propria a 24 bit, la Black Belt Systems offre, ad un prezzo inferiore ai 300$, questo frame buffer in grado di visualizzare fino ad un massimo di 262mila sfumature contemporaneamente (da una palette di 16 milioni). La scheda accetta file in formato IFF-24 e HAM-E, e genera un segnale RGB utilizzabile quindi con un normale monitor Amiga.

ColorBurst: il vantaggio di questo dispositivo a 24 bit della MAST consiste nella compatibilità con qualsiasi modello di Amiga. Da collegarsi tra l'uscita video RGB di Amiga e l'ingresso del monitor, il ColorBurst genera un segnale RGB e consente di miscelare il suo segnale video a 24 bit con quello standard generato da Amiga. La compatibilità video PAL e la presenza di un frame buffer con 1,5 Mb di memoria, unite ad un prezzo di circa 500$, rendono questo prodotto molto appetibile anche per gli utenti europei.

Tramite l'Attribute Requester è possibile salvare su disco i parametri relativi ad una o più facce, per poi richiamarli quando necessario. Dato l'alto numero di variabili presenti, si ottengono ottimi risultati solo in seguito a diversi tentativi e dopo aver fatto molta pratica poiché, nella maggior parte dei casi, non si conoscono a priori le giuste combinazioni per ottenere le superfici volute.

ATTRIBUTE REQUESTER

Abbiamo visto come con l'Attribute Requester, attivabile dopo aver selezionato un oggetto, si definiscano una serie di parametri relativi alla superficie del solido selezionato. È possibile assegnare un nome arbitrario all'oggetto, decidere il tipo di rappresentazione all'interno dell'editor, normale o **Fastdraw** (ovvero sfruttando la *bounding box*), ed applicare alle superfici il **Phong Shading** (un algoritmo veloce di generazione di ombreggiature) per renderle più «morbide».

Il colore in «Imagine» viene definito a 24 bit per pixel: ogni componente colore fondamentale RGB può variare entro un raggio di 255 valori diversi. In modo analogo si definiscono i colori che la superficie deve riflettere (**Reflect Value**) o filtrare (**Transparency Value**), ed il colore dello spot luminoso individuato da una o più sorgenti di luce.

Valori compresi tra 0 e 255 possono essere attribuiti anche agli altri parametri: ad esempio il **Dithering**, che determina la «retinatura» della superficie; il **Roughness** (che ne influenza la rugosità; lo **Shininess**, che definisce la lucentezza, attribuendo alla superficie un aspetto simile alla plastica; l'**Hardness**, che definisce la dimensione dello spot di luce. Nel caso di quest'ultimo parametro, l'uso di valori molto alti determina uno spot decisamente netto e luminoso, simile al classico spot di «**Sculpt-Animate 4D**»; i parametri relativi agli indici di rifrazione infine, vengono utilizzati sugli oggetti che filtrano la luce bianca, deviandola.

TEXTURE E BRUSH

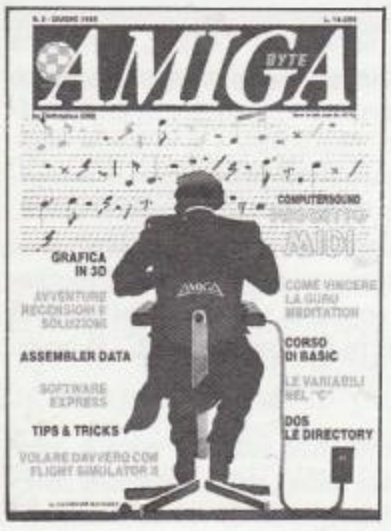
La gestione delle **Texture** viene effettuata attraverso requester specifici, che consentono di impostare tutti i valori necessari per

http://www.oldgamesitalia.net/ http://www.oldgamesitalia.net/

http://www.oldgamesitalia.net/

AMIGA BYTE

SONO DISPONIBILI I FASCICOLI ARRETRATI

(sono già esauriti n. 1-2-3-4-5-6-7-8-11-12-13-22-23 di cui si può avere il disco)

PUOI RICHIEDERE LA TUA COPIA CON DISCO INVIANDO VAGLIA POSTALE DI L. 18.000 AD

AmigaByte, C.so Vitt. Emanuele 15, 20122 Milano.

PER UN RECAPITO PIÙ RAPIDO aggiungi L. 3000 e richiedi SPEDIZIONE ESPRESSO

definire un particolare texture. Per quanto riguarda i **Brush Wrap**, la finestra adibita alla loro gestione ci permette di specificare, oltre che il tipo di *wrapping* (piano od avvolgente), anche la modalità di visualizzazione del brush indicato.

In modo **Color**, la superficie assume i colori dell'immagine IFF che viene importata; i modi **Reflect** e **Filter** rispettivamente riflettono e filtrano le tonalità di colore del brush; il modo **Altitude** permette il bump mapping della superficie, generando effetti di rilievo con una distribuzione dei colori determinata dall'immagine IFF.

Per il posizionamento del brush rispetto alla superficie selezionata si utilizzano i comandi **Edit Axes** e **Transform Axes**.

NOVE TEXTURE DIFFERENTI

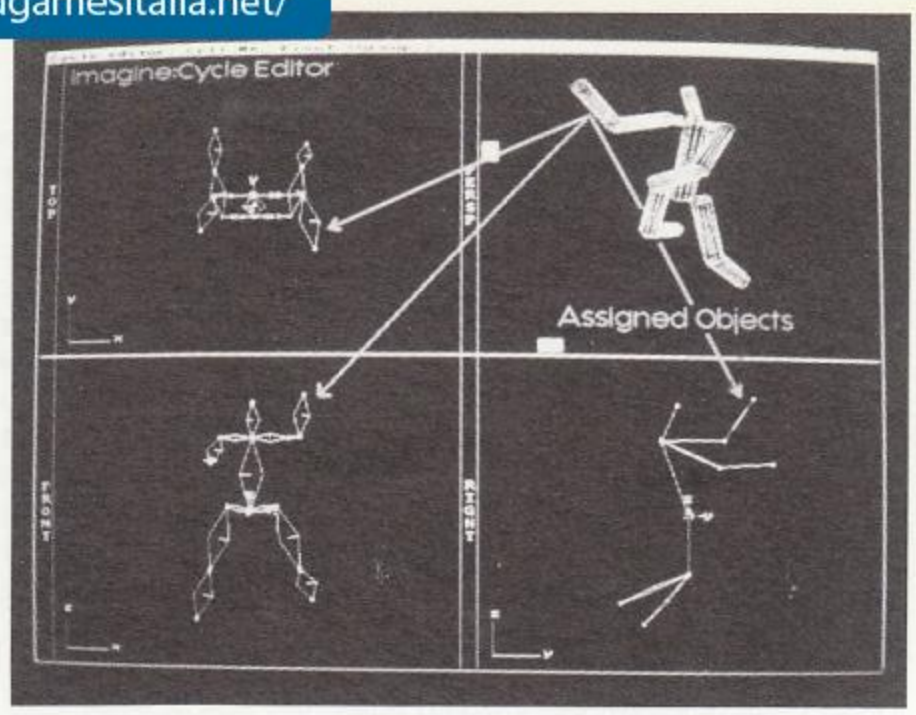
In questa prima versione di «Imagine», sono presenti su disco nove texture differenti. L'immancabile **Checks** genera superfici a scacchiera; **Bricks** simula pareti di mattoni; **Grid** è usato per applicare una griglia ad un solido; **Wood** genera superfici in legno (questa texture comprende anche il **Marble** presente in «Turbo Silver», usato per superfici in marmo); **Disturbed** simula oggetti amorfi ed ondulati; **Dots** distribuisce sul solido una serie di puntini, per ottenere i classici *pois*. Molto potenti sono infine **Angular, Radial** e **Linear** che permettono di modificare il colore di un oggetto lungo determinate direzioni.

FORMS EDITOR

Il **Forms Editor** è stato appositamente inserito nel programma per rendere facile ed intuitiva la creazione di qualsiasi tipo di solido di rotazione. L'operazione ha inizio con il comando **New** che crea una sfera, la più semplice superficie di rotazione, con un numero specificato di punti. È possibile a questo punto deformare le linee di rotazione ed il raggio di curvatura delle linee nel centro.

Le potenzialità offerte sono veramente notevoli ed in molti casi si possono creare solidi difficilmente realizzabili tramite i tradizionali editor: è possibile spostare uno o più punti mantenendo la simmetria frontale e/o laterale, agganciandosi alla griglia di riferimento o utilizzando specifici comandi da menu. In questo editor risulta facilissima la creazione di più oggetti aventi lo stesso numero di punti e forme completamente differenti, e che di conseguenza si prestano in modo particolare ad effetti di *morphing* (metamorfosi) durante una sequenza animata.

Una volta terminato l'editing, è possibile salvare l'oggetto e definirne tutti i parametri di superficie tramite il Detail Editor, od eventualmente apportare qualche ritocco finale. Un solo appunto: con il Forms Editor non è possibile caricare un qualsiasi oggetto e trasformarlo in un *form*, bensì si possono modificare solo solidi di rotazione.



Cycle Editor: per animare una figura complessa si interviene solo sulle linee portanti, senza dover interagire con tutti i punti ed i poligoni presenti.

CYCLE EDITOR

Le difficoltà maggiori che si incontrano nella realizzazione di una sequenza animata riguardano il movimento di figure articolate e complesse, come quella umana. In genere, per questo tipo di animazioni si utilizzano le gerarchie tra solidi, ma la gestione di oggetti di questo tipo risulta ancora molto difficoltosa.

Il **Cycle Editor** rende molto più semplice la realizzazione di una struttura articolata, in quanto lavora unicamente su di uno scheletro della struttura per il movimento delle singole parti. L'utente definisce una struttura «portante» composta da rombi interconnessi tra loro in modo gerarchico e successivamente «assegna» ad ogni singola parte un oggetto complesso creato tramite i due editor di oggetti visti in precedenza. Terminata questa prima operazione, si definiscono tutte le posizioni chiave assunte dall'oggetto lungo un numero predefinito di fotogrammi, lavorando unicamente su semplici strutture visualizzate nelle viste ortogonali dell'editor, e senza dover trattare tutte le superfici ed i punti che compongono

http://www.oldgamesitalia.net/
http://www.oldgamesitalia.net/

http://www.oldgamesitalia.net/

effettivamente l'oggetto. L'oggetto completo viene visualizzato nella quarta finestra, quella della vista prospettica. Il **Cycle Object** risultante equivale ad una sorta di AnimBrush tridimensionale, che contiene le informazioni relative a tutte le posizioni assunte entro predefiniti intervalli di tempo. Un Cycle Object viene animato dal programma come fosse un qualsiasi oggetto «semplice».

STAGE EDITOR

Terminata la fase di editing degli oggetti si passa allo **Stage Editor**, tramite il quale si stabiliscono le posizioni della telecamera, dei solidi e delle sorgenti di luce, per procedere successivamente al rendering finale.

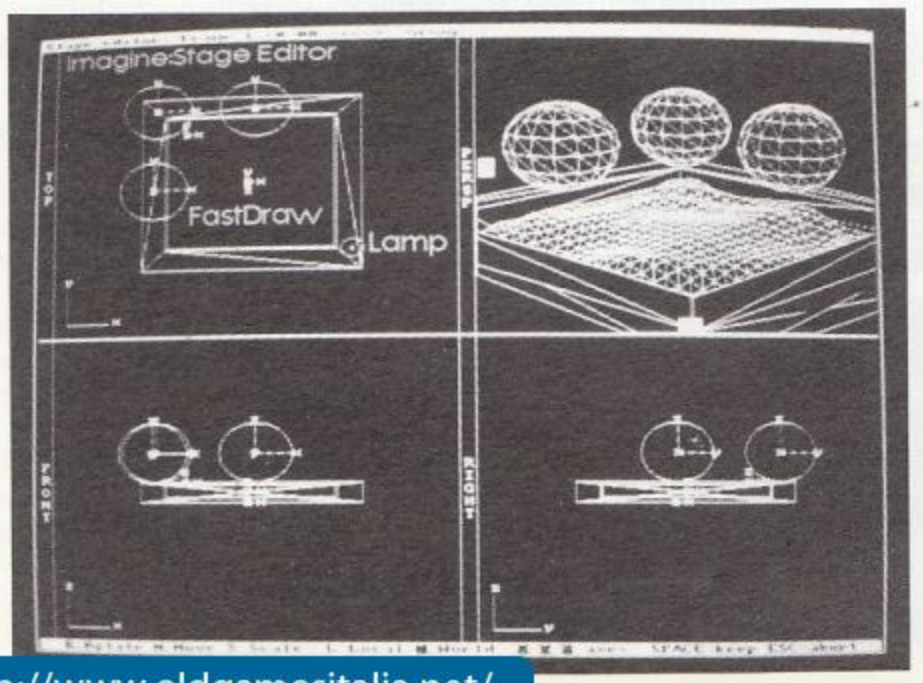
L'ambiente di lavoro è simile a quello già incontrato nel Detail Editor: gli oggetti vengono caricati e posizionati con gli stessi strumenti. È possibile cambiare la dimensione dei singoli oggetti, ma tutte le altre caratteristiche già definite in precedenza, come il tipo di superficie, non possono essere modificate. La telecamera ed eventuali sorgenti luminose vengono trattate come fossero oggetti, anche se con le dovute limitazioni.

La creazione di **path** (linee guida per il movimento di uno o più oggetti) viene gestita da uno specifico menu. Ogni path è rappresentato da una curva di Bezier, che viene ricalcolata in tempo reale ogni volta che si effettuano spostamenti o rotazioni sui punti chiave della linea. i Path possono essere aperti o chiusi, e possono assumere praticamente qualsiasi forma.

La grande potenza dello Stage Editor risiede nella **TimeLine**, attivata dal comando **Action**. In questo ambiente si definiscono tutte le specifiche di una sequenza animata, come il numero di fotogrammi, le posizioni degli oggetti, della telecamera o dei punti luce. «Imagine» tratta gli oggetti assegnando loro una serie di specifiche che influiscono sulla posizione, sull'orientamento, sulla forma e sulla dimensione del solido stesso. Ogni entità (oggetto semplice, Cycle Object, telecamera o punto luce) viene definita come **Actor**, con un proprio nome. Ad un Actor possiamo far corrispondere differenti oggetti lungo la TimeLine per effettuare metamorfosi in un numero definito di fotogrammi.

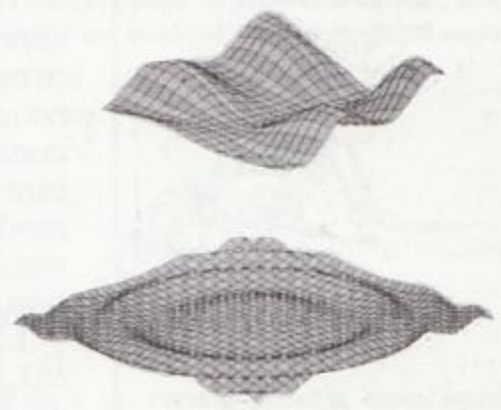
Va ricordato che si possono modificare non solo la forma di un oggetto, ma anche tutti i parametri relativi quali il colore, il tipo di

Tramite lo Stage Editor si posizionano gli oggetti, le sorgenti di luce, e la telecamera per il rendering finale.



WAVES 2.1

Uno straordinario tool grafico indispensabile per la realizzazione di suggestivi effetti animati!



WAVES genera i dati necessari alla creazione di animazioni di onde tridimensionali con DELUXE PAINT III, SCULPT ANIMATE 4D, VIDEOSCAPE 3D ed altri diffusi pacchetti grafici.

È richiesto preferibilmente almeno 1 MB di memoria.

Per ricevere i due dischetti di WAVES 2.1 invia vaglia postale ordinario di lire 49.000 ad AmigaByte, C.so Vitt. Emanuele 15, Milano 20122.

Specifica sul vaglia stesso la tua richiesta ed il tuo indirizzo. Per un recapito più rapido, aggiungi lire 3.000 e richiedi la spedizione espresso!

http://www.oldgamesitalia.net/
http://www.oldgamesitalia.net/

http://www.oldgamesitalia.net/

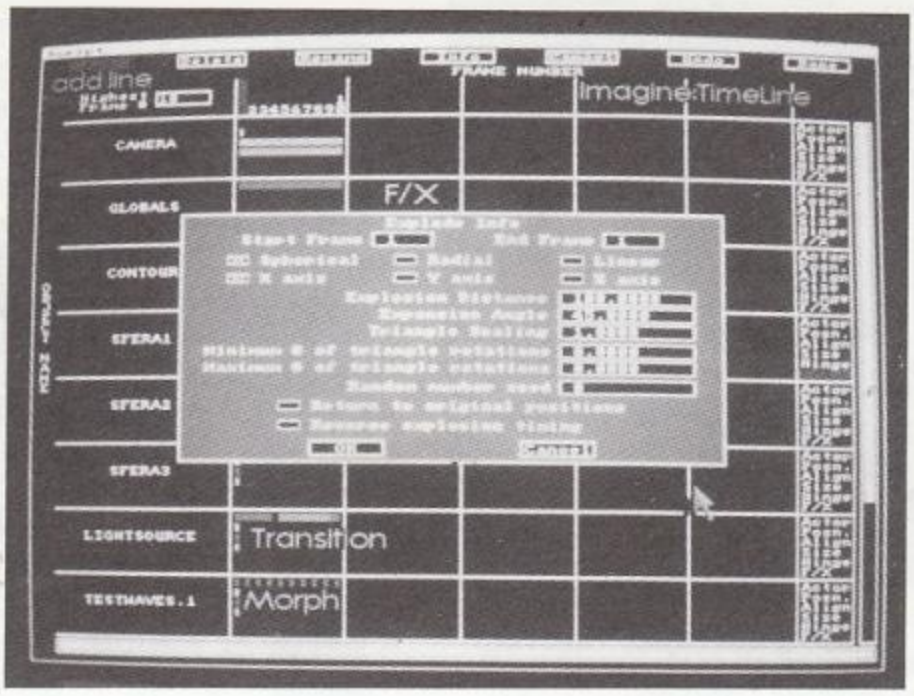
C PACKAGE

5 DISCHI!

C MANUAL: Un corso completo di programmazione in C. Dieci capitoli ed oltre settanta esempi, con sorgenti commentati e già compilati. 3 DISCHETTI.
ZC: Un pacchetto di sviluppo completo di compilatore, assemblatore, ottimizzatore, linker e librerie, per produrre eseguibili perfettamente funzionanti. 1 DISCHETTO.
GWIN: Una vasta raccolta di funzioni per rendere semplice ed intuitiva la gestione di schermi, finestre ed il tracciamento di grafica. 1 DISCHETTO.
NB: Tutta la documentazione è in inglese.

✻

Per ricevere i dischetti di C Package invia vaglia postale ordinario ad AmigaByte, C.so Vitt. Emanuele 15, Milano 20122. Lire 10.000 ogni singolo dischetto (o lire 40.000 tutti e cinque). Specifica sul vaglia stesso la tua richiesta ed il tuo indirizzo. Per un recapito più rapido, aggiungi lire 3.000 e richiedi la spedizione espresso!



Il requester dell'effetto speciale Explosion; sullo sfondo, si intravede lo schermo della Time Line.

superficie ed i texture presenti. Questi ultimi sono di tipo dinamico, e di conseguenza possono subire qualsiasi tipo di modifica durante un'animazione (**Texture Animation**).

La capacità di variare, nel corso di un'animazione, non soltanto la forma di un oggetto ma anche il suo aspetto ed il materiale di cui è composto, si presta alla realizzazione di effetti davvero suggestivi ed impossibili da creare con altri pacchetti grafici.

La posizione di ogni Actor può essere specificata lungo la sequenza secondo **Key Frames** oppure secondo un path predefinito, presente a sua volta come Actor nella sequenza. Per movimenti complessi (ad esempio i moti orbitali), è possibile agganciare un path ad un oggetto, che a sua volta si sta muovendo lungo un altro path. L'allineamento (ovvero l'orientamento degli assi relativo rispetto al sistema di riferimento assoluto) viene determinato in base a tre criteri: per posizioni chiave (Key Frames), per allineamento ad un path, e per *tracking* di un altro Actor (in questo modo la telecamera può seguire e «guardare» un oggetto in movimento); la dimensione di un oggetto viene invece definita unicamente per Key Frames.

Per particolari effetti, possiamo far «orbitare» un Actor attorno ad un altro oggetto, utilizzando la linea chiamata **Hinge**.

Sono disponibili inoltre i cosiddetti **F/X** (effetti speciali), quali Explosion, Ripple e Grow.

Gli effetti speciali sono, probabilmente, una delle peculiarità più entusiasmanti di «Imagine»: con **Explosion** possiamo infatti far «esplodere» un oggetto in tutti i suoi componenti base (triangoli) con notevole realismo o, tramite lo stesso F/X, è possibile ottenere l'effetto inverso di implosione e di dissolvenza di un solido.

Con **Ripple** possiamo simulare una superficie fluida perturbata da un corpo, ottenendo onde molto realistiche; per realizzare effetti di questo tipo esiste inoltre un'utility dedicata, «**Waves 2.1**», ancora più versatile e potente, recensita sul fascicolo 29 di AmigaByte.

Tramite l'effetto **Grow**, infine, si può estrudere una linea piana lungo una sequenza ottenendo risultati davvero spettacolari. È da sottolineare che gli F/X, così come i Texture, sono memorizzati su disco separatamente dal programma principale: è lecito prevedere quindi che in breve tempo saranno disponibili e verranno commercializzati nuovi effetti e Texture aggiuntivi.

ATTUALMENTE IL MIGLIORE

«Imagine» è senza dubbio il programma 3D più completo attualmente disponibile per Amiga, e in questa sua prima release 1.0 è risultato abbastanza stabile e funzionale, sebbene qua e là faccia la sua comparsa qualche occasionale bug. La qualità delle immagini generate da «Imagine» si può apprezzare pienamente solo utilizzando hardware adeguato, ovvero schede grafiche come la «FireCracker 24», in grado di visualizzare 16.7 milioni di tinte contemporanee; come software di grafica a 24 bit, «Imagine» non ha concorrenti nel mondo Amiga e le sue prestazioni possono essere tranquillamente paragonate a quelle di pacchetti ben più costosi e diffusi in ambito professionale su altri computer.

□

http://www.oldgamesitalia.net/
http://www.oldgamesitalia.net/

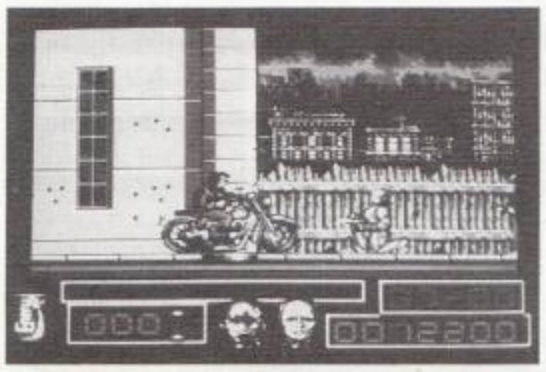
http://www.oldgamesitalia.net/

Tips & Tricks

SUGGERIMENTI E TRUCCHI VARI

Un trucco per il vecchio gioco arcade con grafica vettoriale «Virus»: tenete premuto il tasto Enter sul tastierino numerico e premete i tasti P ed O. Attiverete un cheat mode che vi consentirà di accedere ad alcune funzioni speciali, premendo i tasti F (rifornimento carburante), C (effetti speciali) ed L (spostamento linee). Con il tasto N potrete riprendere il gioco nel punto in cui si era interrotto.

Uno tra i giochi più impegnativi apparsi negli ultimi tempi è «Robocop 2» della Ocean: senza il cheat-mode e le sue provvidenziali vite infinite, difficilmente si riesce



a portarlo a termine senza soccombere ai colpi d'arma da fuoco dei delinquenti che infestano tutti i livelli di gioco.

Per fortuna, è sufficiente digitare le parole magiche SERIAL INTERFACE per far diventare immortale Robocop.

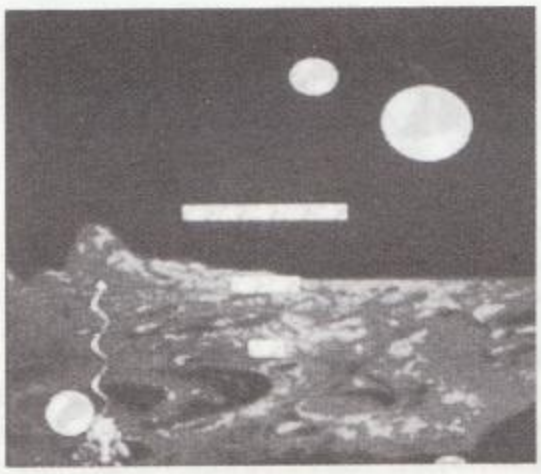
Quando appare la schermata con il titolo del programma nel gioco «Drivin' Force» della Digital Magic, clickate sulle due «I» della parola «Drivin'»: durante la corsa l'auto non uscirà mai di strada.

In «Toyota Celica Rally» della Gremlin Graphics potete invece premere contemporaneamente i tasti «Ctrl» e «C» all'inizio della corsa, per bloccare il cronometro.

Infine, in «Nitro» della Psygnosis, inserire il nome MAJ come sigla del giocatore ed otterrete in cambio 5000 unità di fuel e 50 crediti.

Alcuni codici per accedere ai livelli avanzati di «Oops Up!», il diabolico gioco delle sfere rimbalzanti ispirato all'arcade «Pang».

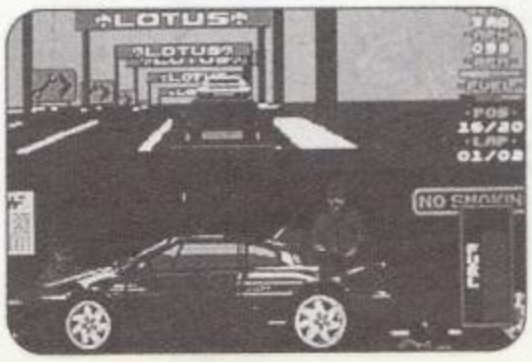
Livello 10: D04G
Livello 20: DK49



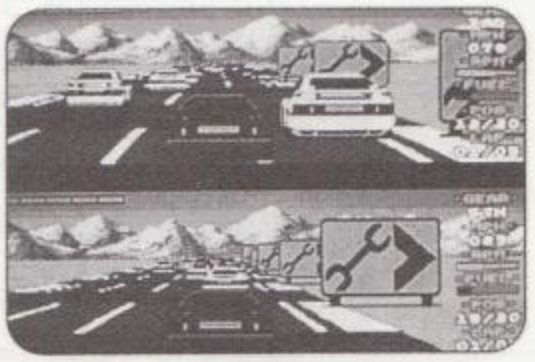
Livello 30: 10F4
Livello 40: 4G7H
Livello 50: S04L
Livello 60: MC90
Livello 70: FUK0
Livello 80: EB01
Livello 90: A234
Livello 100: 4799

Nell'arcade-adventure «Rings of Medusa» il cheat mode si attiva all'inizio del gioco digitando la parola DESOXYRIBONUKLEINSAEURE e premendo il tasto Help dopo aver inserito il proprio nominativo.

Attualmente, tra i numerosi giochi di corsa automobilistica in circolazione, uno tra i più avvincenti e giocabili è «Lotus Esprit Turbo Challenge» della Gremlin Graphics. Anch'esso, come di regola, contiene un cheat mode nascosto che permette ai giocatori di qualificarsi per la corsa anche senza bisogno di ottenere tempi strepitosi.



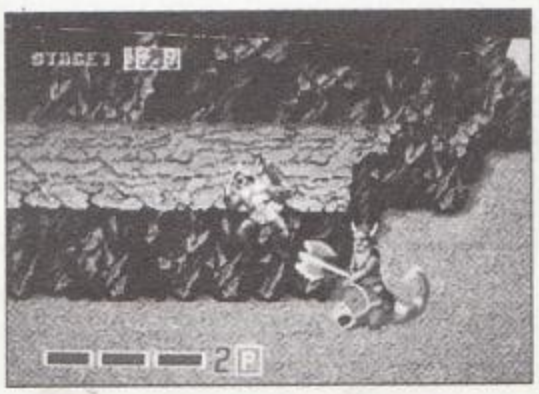
È sufficiente selezionare il modo a due

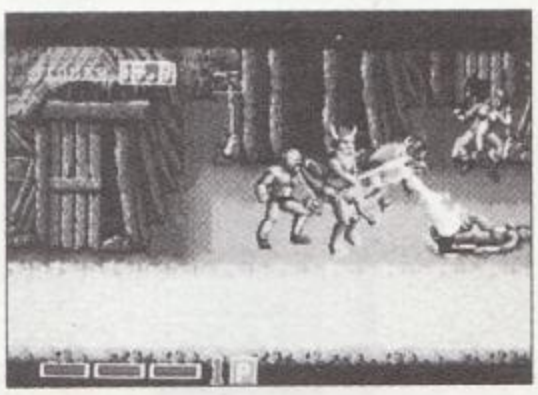


giocatori ed inserire, al posto dei nominativi dei due piloti, queste due frasi: IN A BIG COUNTRY per il primo giocatore e FIELDS OF FIRE per il secondo.

Pare inoltre che digitando invece MONSTER come nome del primo pilota e SEVENTEEN come nome del secondo, venga attivato l'accesso ad un house-game nascosto.

Non è un vero e proprio cheat-mode, ma può risultare ugualmente utilissimo questo trucco in «Golden Axe» della Sega: quando state per morire, durante una partita ad un solo giocatore, premete il tasto del mouse o del joystick inserito nell'altra porta del computer. In questo modo farete entrare in gioco un altro personaggio, che partirà con tutte le vite e l'energia





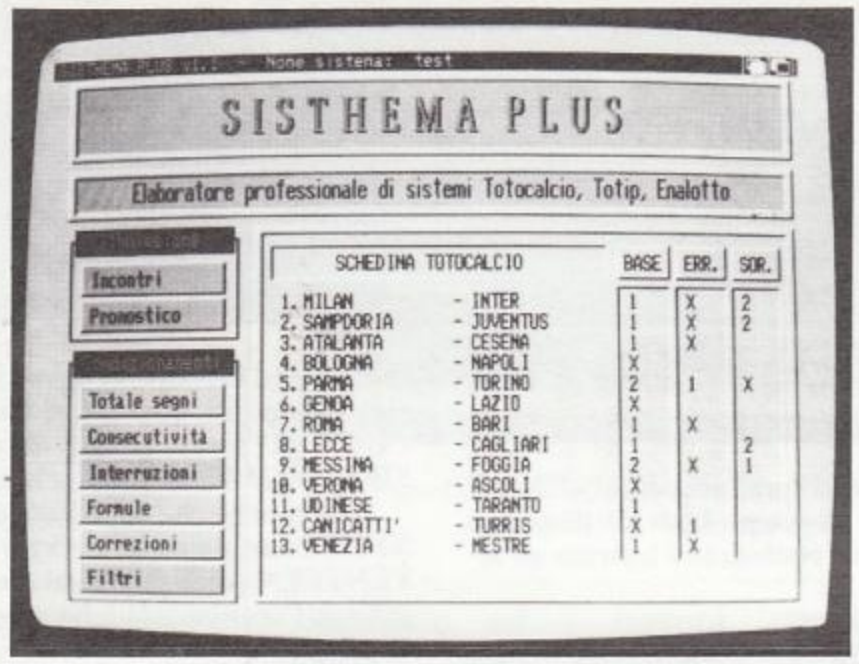
completa a disposizione, e potrete proseguire la partita anche dopo la morte del vostro primo guerriero.

http://www.oldgamesitalia.net/
http://www.oldgamesitalia.net/

http://www.oldgamesitalia.net/

TOOLS

a cura di Domenico Pavone



SISTHEMA PLUS

Per un sogno tipicamente italiano, un programma tipicamente italiano. Non potrebbe definirsi altrimenti questo prodotto della Progetto Software, il cui nome porta già in sé le speranze dei molti aspiranti «tredicisti» della domenica. In effetti, per una più corretta definizione, andrebbero anche citati i «dodicisti» del sabato, in quanto «**Sisthema Plus**» è un tool dedicato allo sviluppo di sistemi ridotti e condizionati finalizzati ai concorsi Totocalcio, Totip ed Enalotto.

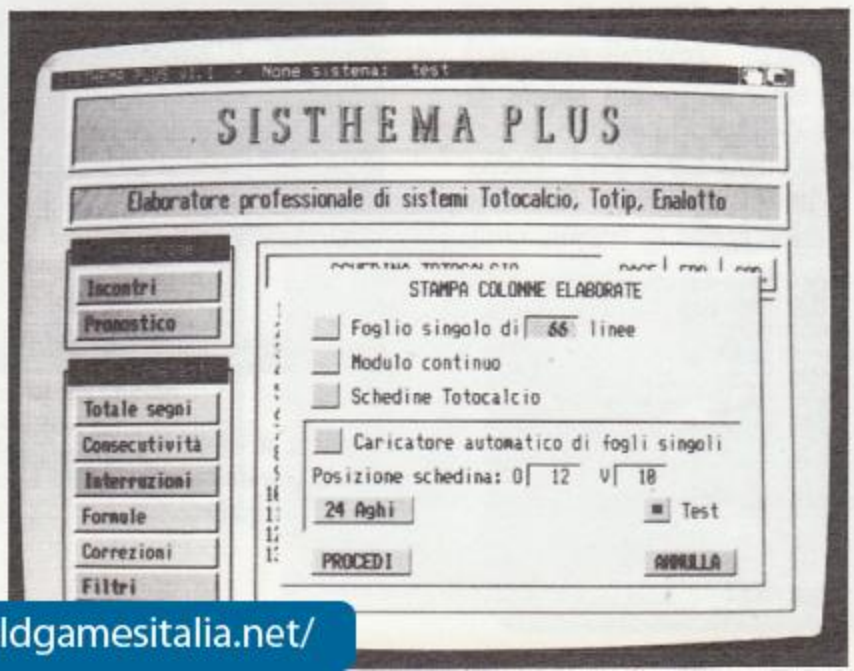
Con l'avvento della cosiddetta informatizzazione di massa, programmi del genere hanno cominciato a fiorire un po' per tutti i modelli di computer, ed Amiga non poteva certo fare eccezione. «Sisthema Plus», pur senza esibire nulla di nuovo in un settore (quello della sistemistica) divenuto altamente specializzato, propone tuttavia un approccio molto lineare e soprattutto semplice da gestire, sposando appieno la filosofia Intuition-oriented di Amiga. Per di più, ed è forse questa la caratteristica in grado di fare la differenza, consente non solo la stampa degli elaborati su tabulato cartaceo, ma anche la compilazione fisica delle schedine. Il che, se si pensa che taluni sviluppi possono comprendere migliaia di colonne, rappresenta un notevole motivo di interesse per i «professionisti» del botteghino. Questo tipo di stampa, particolarmente critico per la precisione necessaria, è inoltre facilitato dalla possibilità di adottare caricatori hardware automatici, o di effettuare microregolazioni software e test di stampa che includono la scelta tra stampanti a 9 e 24 aghi.

Dello stesso programma esiste anche una versione più economica, «Sisthema», che si differenzia dalla «Plus» solo per l'assenza delle opzioni di stampa su schedina, mentre entrambe richiedono almeno 1 Megabyte di Ram per operare senza problemi.

L'ambiente di lavoro è caratterizzato da un unico schermo dal quale si accede a pochi e semplici menu a tendina, mentre tutte le opzioni che riguardano lo sviluppo di un sistema, o meglio il suo condizionamento, sono affidate ad una serie di otto riquadri clickabili con il mouse. Per ogni opzione viene aperta una nuova finestra per gli input necessari, anch'essi tutti inseribili sfruttando il solo mouse. Nella sezione centrale dello schermo campeggia invece la schedina vera e propria, ovvero la maschera che conterrà la specifica degli incontri ed il pronostico del giocatore. Questi dati, dopo aver selezionato la relativa icona («Incontri» e «Pronostico»), andranno ovviamente immessi da tastiera.

Per il pronostico sono presenti tre

settori, nei quali è possibile inserire il segno desiderato (non ci crederete, ma ci si sta riferendo ai simboli 1, X, 2) in base ad una gerarchia di valori: il pronostico di base, quello eventualmente passibile di errore, e la «sorpresa». Questa suddivisione, come ovvio, riguarderà unicamente i sistemi ridotti e/o condizionati, principale campo d'azione di «Sisthema Plus». Quelli integrali incondizionati, infatti, pur se trattati dal programma, è chiaro che lo sono in funzione più che altro informativa (numero di colonne risultanti, loro



http://www.oldgamesitalia.net/
http://www.oldgamesitalia.net/

http://www.oldgamesitalia.net/

costo in soldoni, eccetera): per questi, è sufficiente compilare una sola schedina quale che sia il numero di colonne risultanti.

Una volta immesso il pronostico, si possono considerare due fasi: quella del condizionamento e quella della riduzione. Quest'ultima, come ovvio, è totalmente a carico del software, che produrrà un numero di colonne da giocare (il cosiddetto «sviluppo») proporzionale al numero di «doppie» e «triple» presenti nel pronostico, al tipo di riduzione adottata (Semplice oppure Ottimizzata, dal menu Elaborazione), ed ai condizionamenti imposti. Una doppia scelta («normale» o «veloce») è consentita anche per lo sviluppo di sistemi integrali condizionati. Senza troppo entrare nel merito di un argomento ben noto a tutti i sistemisti, i condizionamenti implementati sono sei, e precisamente:

Totale Segni: per limitare ad un massimo di libera scelta la ripetizione di uno stesso simbolo in una singola colonna. Per fare un esempio, è di solito molto improbabile, quantomeno nel Totocalcio, che una colonna vincente contenga più di tre o quattro segni «2».

Consecutività: in questo caso, la riduzione non prenderà in considerazione il ripertersi di uno stesso segno per più di un certo numero di volte consecutive. Statisticamente, solo in rari casi si ha una sequenza di tre o quattro risultati consecutivi con simbolo «1», ed ancora più rara l'eventualità di vedere più di due simboli «2» consecutivi.

Interruzioni: in pratica, il concetto opposto di consecutività.

Formule: un po' più complicato da capire, questo condizionamento prevede l'esame statistico di quanto avvenuto in precedenti concorsi, prendendo in esame la frequenza con la quale i tre segni si sono succeduti. In base a tali calcoli, anch'essi peraltro noti agli specialisti della schedina, vengono prese in considerazione delle «terzine» che condizionino il rapporto di incidenza dei tre simboli.

Correzioni: per modificare l'influenza del numero di errori o delle «sorprese» sulla riduzione.

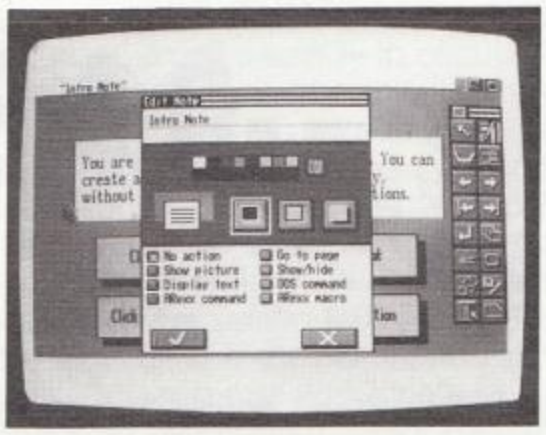
Filtri: ulteriore sofisticazione del concetto di «totale segni».

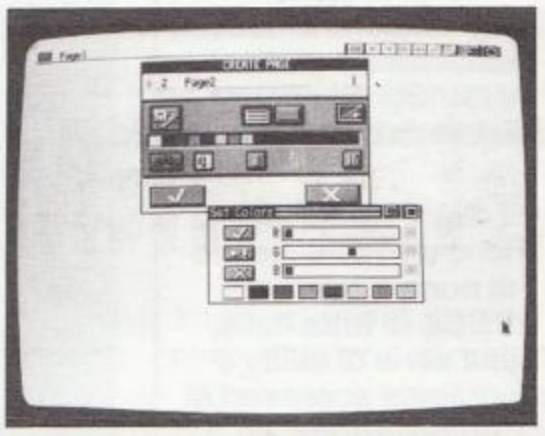
Una volta impostati (o meno) questi parametri, si può poi ordinare al programma di iniziare la riduzione, che richiederà tempi proporzionali al numero di colonne da esaminare, ed alla complessità dei condizionamenti imposti. Tutte le informazioni sulle caratteristiche assegnate ad un sistema possono naturalmente essere salvate su periferica, e richiamate in un secondo tempo per essere applicate ad altri pronostici.

Già detto delle notevoli prestazioni in fase di stampa, non resta da segnalare che l'opzione di controllo sulle eventuali colonne vincenti: in un sistema ridotto di grosse proporzioni, queste potrebbero celarsi tra migliaia di altre, con intuibili difficoltà di reperimento. Un'opzione molto comoda, quindi, in grado di evitare al sistemista ulteriori fatiche, che potrebbero però essere anche ben accette. Nel caso di una grossa vincita, naturalmente...

PROGETTO SOFTWARE
Via Rodi, 39
10095 - Grugliasco (TO)

HYPERBOOK





L'associazione di termini come «professional» al più svariato e complesso software di DeskTop Publishing/Video, o meramente orientato alla grafica o, ancora, volto alle più o meno mirabolanti potenzialità del «multimediale», può facilmente indurre un utente di Amiga ad errate valutazioni nei confronti di un programma come «**HyperBook**». E questo non tanto per le sue qualità intrinseche, ma perché esso rappresenta una (rara) via di mezzo tra il professionale e l'amatoriale che tuttavia consente, con una facilità sbalorditiva, realizzazioni solitamente riservate ad un'utenza molto più specializzata. Tentando una definizione stringata, «HyperBook» è un tool che permette di organizzare in un unico contesto elementi di diversa natura quali possono essere grafica, testo, o altri programmi eseguibili di Amiga. Il tutto con notevoli capacità interattive, grazie alle quali è quindi possibile creare ogni sorta di demo «multimediali» quali ad esempio tutorial didattici o presentazioni dimostrative, ma anche (perché no?) storie illustrate, manuali, o addirittura particolari database in cui ogni «pagina» può far riferimento ad informazioni grafico/testuali.

In pratica si potrebbe considerare «HyperBook» alla stregua di una sorta di linguaggio di programmazione, in cui però tutte le componenti sono gestibili con estrema semplicità a suon di mouse, e che sfrutta come termini propri un limitato numero di «oggetti» ad ognuno dei quali è possibile associare un certo tipo di azione. I più esperti potranno addirittura attivare, tramite gli oggetti, delle macro Arexx che coordinino l'attività di più programmi esterni.

Entrando nel vivo dell'argomento, questi oggetti sono distinti in cinque tipi: Buttons, Notes, Drawings, Pictures e Lists. La realizzazione di un progetto qualsiasi prevede in pratica la creazione di un «hyperbook», composto da una o più «pagine», in ognuna delle quali è possibile posizionare la serie di «oggetti» che più aggrada. A lavori conclusi, l'hyperbook potrà essere consultato e sfogliato tramite semplici click

http://www.oldgamesitalia.net/
http://www.oldgamesitalia.net/

http://www.oldgamesitalia.net/

MODEM DISK

Tutto il miglior software PD per collegarsi a banche dati e BBS e prelevare gratuitamente file e programmi!

☆

Un programma di comunicazione adatto a qualsiasi modem, dotato di protocollo di trasmissione Zmodem, emulazione grafica ANSI/IBM ed agenda telefonica incorporata.

☆

Il disco comprende anche un vasto elenco di numeri telefonici di BBS di tutta Italia, una serie di utility e programmi accessori di archiviazione, ed istruzioni chiare e dettagliate in italiano su come usare un modem per collegarsi ad una BBS e prelevare programmi.

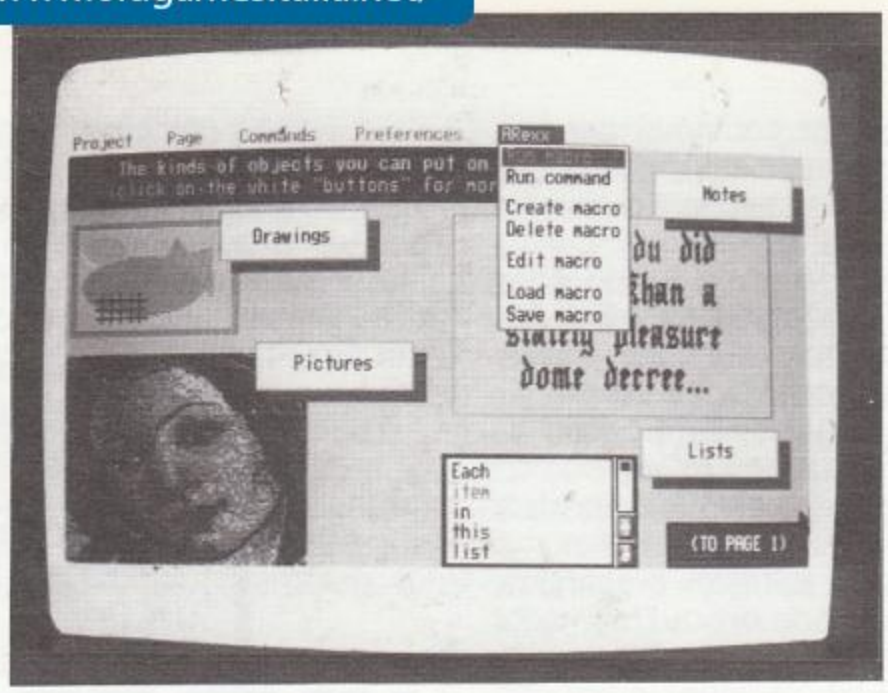
☆

Per ricevere il dischetto MODEM DISK invia vaglia postale ordinario di lire 15.000 ad AmigaByte, C.so Vitt. Emanuele 15, Milano 20122.

☆

Specifica sul vaglia stesso la tua richiesta ed il tuo indirizzo. Per un recapito più rapido, aggiungi lire 3.000 e richiedi la spedizione espresso!

BBS 2000
24 ore su 24
02-76.00.68.57
02-76.00.63.29
300-1200-2400
9600-19200 BAUD



del mouse.

Ad un semplice e lineare utilizzo del prodotto finale corrisponde un altrettanto semplice (per non dire divertente) ed intuitivo uso del programma in fase di creazione. L'ambiente di lavoro è rappresentato da uno schermo che ricorda vagamente quello di «Deluxe Paint», con una serie di riquadri (iconizzabili) posti sul lato destro del video, tramite i quali si può praticamente fare tutto. Molte altre opzioni, comunque, sono presenti nella title bar dello schermo, tutte di immediata comprensione. Per ottimizzare al massimo il multitasking, il problema prevede inoltre una sua possibile «iconizzazione», riattivabile clickando nel gadget posto a sinistra della mini-finestra residua.

L'attività principale consiste nell'assegnare i giusti compiti agli «oggetti», e nel rendere graficamente accettabile l'insieme. Clickando nelle apposite icone si potrà posizionare, e contemporaneamente dimensionare, un oggetto in qualunque punto del monitor. Ciò fatto, a seconda del tipo di oggetto prescelto, una nuova finestra (attivabile in qualsiasi momento con un click del pulsante destro del mouse, dopo avere selezionato la prima icona in alto a destra dello schermo) consentirà di assegnare ad esso una certa azione. Questa potrà essere la visualizzazione di un testo, di un'immagine IFF a tutto schermo, o di un clip grafico «ritagliabile» da una schermata. Oppure ancora, l'azione potrà far cambiare pagina, eseguire una istruzione Dos, o attivare un comando (o una macro) di ARexx. Per lo più, simili azioni sono assegnate ad oggetti di tipo «button», ma la stessa cosa può essere fatta con oggetti «text», nei quali è intuibilmente inseribile qualunque testo redatto con il font che più aggrada. Analogamente, un oggetto Drawing potrà essere disegnato grazie ad una serie di tool interni ad «HyperBook», che poco hanno da invidiare alle più blasonate utility grafiche (cerchi, linee, poligoni, brush diversi, e così via). Il «drawing», tra l'altro, implementa una grafica strutturata, non bitmap, che può quindi essere poi facilmente modificata nelle dimensioni senza minimamente perdere in qualità ed in definizione.

Altra interessante opzione, quella che consente di caricare un'immagine Iff esterna, e di prelevarne una sezione (delimitandola con il mouse), da sovrapporre poi nello schermo che costituisce la «pagina» corrente dell'hyperbook in fase di gestazione. Anche l'immagine IFF, per la cronaca, può essere ridimensionata senza problemi una volta importata nell'hyperbook.

Scelto il tipo di azione da associare all'oggetto, un nuovo requester mostrerà poi i device disponibili dai quali dovrà essere prelevato l'eventuale file da esibire, se si è optato per una visualizzazione. Il tutto potrà sembrare complesso, ma in realtà è più difficile descriverlo che non utilizzarlo: bastano pochi minuti per impadronirsi del programma, prima ancora di aver consultato il manuale.

Dagli «oggetti» disponibili, il più anomalo (ma anche il più interessante) è quello di tipo «List». Suo tramite, si ha la possibilità di creare un vero e proprio requester, sullo stile di quanto si è abituati a vedere nei programmi che richiedono un file in input, ma le cui «entry» diverranno in realtà un menu. Posizionato l'oggetto con la consueta manovra del mouse, si può infatti digitare in ogni sua riga un testo di nostro gradimento, quindi associargli una certa azione. Ma si badi: stavolta l'a-

http://www.oldgamesitalia.net/
http://www.oldgamesitalia.net/

http://www.oldgamesitalia.net/

zione riguarderà non l'oggetto in toto, ma la riga del requester che verrà clickata nell'uso finale! Tradotto in parole povere, si potrebbero per esempio inserire nel finto-requester le voci di un ipotetico archivio di qualsiasi genere, con un file di testo o grafico memorizzato su disco che verrà mostrato quando ne verrà selezionato il nome con il mouse. Nulla vieta, comunque, che anche in questo caso si associ un altro tipo di azione, similmente a quanto possibile con gli altri «oggetti». Ogni voce inserita nel «listing» può poi essere caratterizzata da un diverso colore, per una più chiara demarcazione visiva degli (eventuali) compiti ad essa collegati.

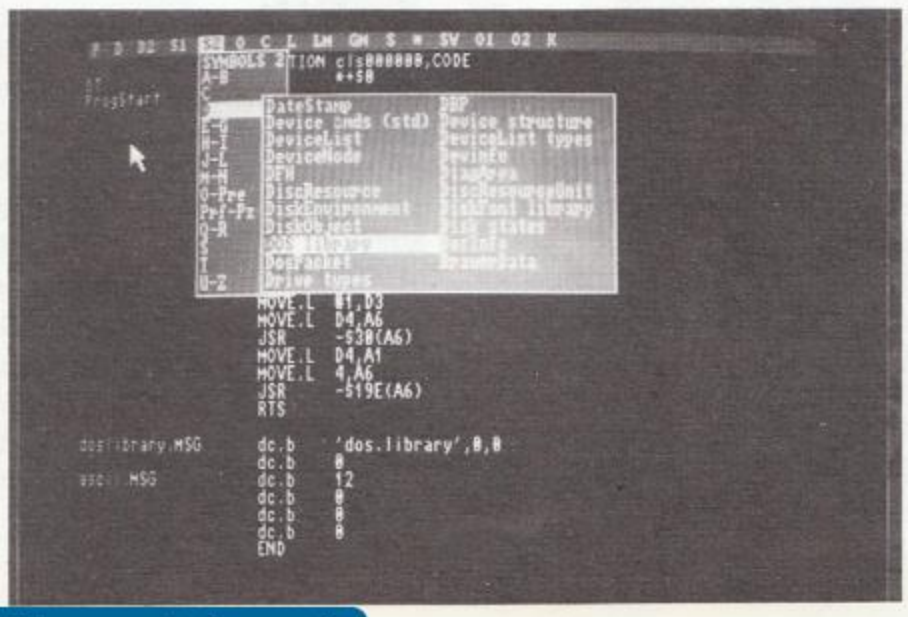
Il package di «HyperBook» comprende inoltre un editor di testi autonomo, ben provvisto di tutto il necessario e, particolarmente, due demo da provare prima di ogni altra cosa. Ci si renderà conto così di cosa è in grado di fare il programma, soprattutto se utilizzato con raziocinio. Se infatti è relativamente semplice mettere insieme un mosaico fatto di programmi, testi, grafica, e chi più ne ha più ne metta, è anche vero che per giungere ad un risultato degno di nota la fase «progettuale» riveste un'importanza non secondaria. E questa, inutile dire, non fa parte del corredo di «HyperBook».

GOLD DISK INC.
P.O. Box 789, Streetsville
Mississauga, Ontario
Canada L5M 2C2

RESOURCE 4.0

Nato come semplice disassemblatore interattivo per il linguaggio macchina, «**Resource**» si è costantemente evoluto fino a raggiungere, con l'attuale versione 4.0, dimensioni e caratteristiche tali da non rendergli giustizia restringendone il campo d'azione in questa categoria. Unico altro esempio del genere, con le dovute differenze, può infatti essere considerato il programma «DSM» (OTG Software), mentre risulta privo di fondamento ogni accostamento con i vari «NewZap», «Filemaster», e software analogo. Al pari di questi, «Resource» può sì consentire la modifica di un file preesistente, ma ad un livello decisamente diverso, che richiede come condizione indispensabile una certa dimestichezza con la programmazione in Assembly.

Scopo principale di «Resource» è quello di automatizzare al massimo una procedura che, da un semplice disassemblato simile a quello ottenibile ricorrendo ad un qualunque programma monitor, giunga alla più completa e chiara espressione simbolica dello stesso. In pratica, permette lo studio approfondito di qualunque programma, ma anche una sua modifica sostanziale, in quanto il prodotto finale delle manipolazioni consentite può poi essere facilmente riassemblato ricorrendo ad uno dei tanti package riservati alla programmazione Assembly. Lo stesso «Resource» permette addirittura di preselezionare da menu (Options2/Assembler) la compatibilità con tre diversi pacchetti (Metacomco, Cape e Capatch), ma di fatto è poi possibile adottare un qualunque formato, grazie alla marea di opzioni offerte dal programma.



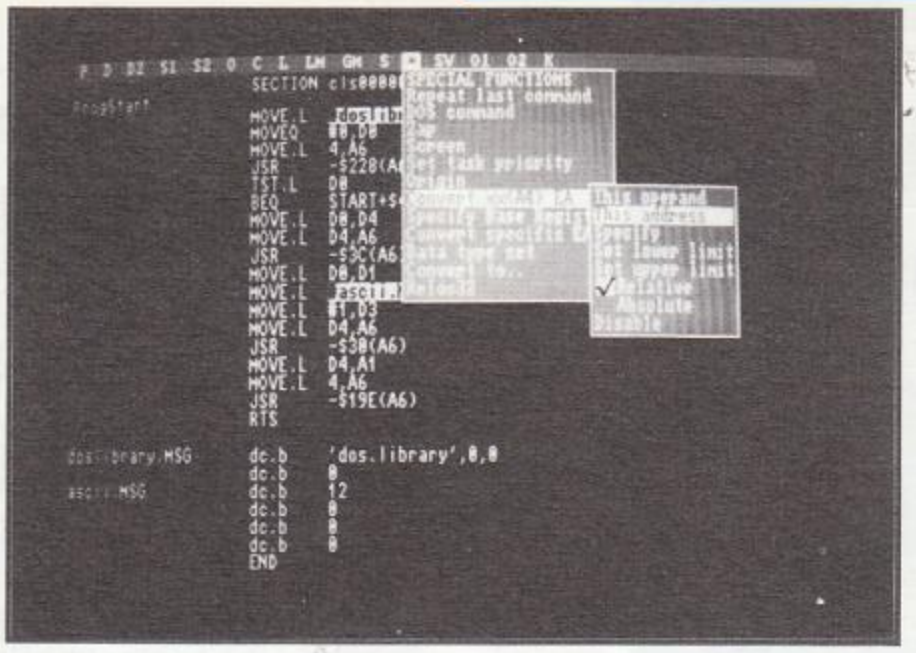
Il ricorso ad un assemblatore esterno non è comunque obbligato, soprattutto se si adopera «Resource» per apportare minime modifiche all'input originario: anche dall'interno del suo ambiente di lavoro è infatti possibile modificare, per esempio, alcuni codici, e salvare poi il tutto con l'opzione Save/Binary image; l'output sarà in questo caso un file eseguibile. In alternativa, può essere adottata l'opzione Save.asm, che produce invece un file ascii con il disassemblato completo di quanto elaborato da «Resource». Come ovvio, questo può poi essere trattato con un qualunque editor di testi, e riassemblato assieme alle modifiche apportate in questa fase. Ma procediamo con ordine, chiarendo meglio le potenzialità del programma, a cominciare dall'input richiesto.

Fin qui, infatti, si è detto genericamente che è in grado di trattare qualunque «file», ma in realtà «Resource» può fare molto di più. Adottando l'opportuna sintassi da ambiente Dos («Resource» non può essere attivato da WorkBench), o molto più comodamente ricorrendo ai menu a tendina dopo che si è lanciato il programma, si può scegliere se disassemblare un file, una porzione della memoria di Amiga (Project/Disassemble Memory), o anche tracce e settori di un disco (Project/Read Tracks).

Nel caso dei file, sono inoltre possibili due forme di caricamento: tramite l'opzione Open Load File, sempre del menu Project, che prevede una totale rilocazione dello stesso, esattamente come se fosse stato lanciato per essere eseguito. In questo caso si avrà sullo schermo un disassemblato per così dire «tradizionale», limitato alla sequenza dei codici mnemonici dell'Assembly che individuano il file in esame. Scegliendo invece l'opzione Open Binary File, si avrà accesso anche a tutta la serie di informazioni trattate da Amiga in modo «trasparente» (normalmente non visibili), ma che vengono memorizzate insieme ad un file: l'header, gli hunk, eventuali simboli per il debug, e co-

http://www.oldgamesitalia.net/
http://www.oldgamesitalia.net/

http://www.oldgamesitalia.net/



sì via. Salvo casi particolari, sarà comunque «**Load File**» l'opzione più adoperata, in quanto consente un accesso semplificato non solo ai programmi eseguibili, ma anche per esempio a librerie o device driver, che possono quindi anch'essi essere disassemblati, salvati in forma di codice sorgente, modificati, ed infine riassemblati.

Già dopo il caricamento iniziale di un file, semplificato dall'adozione dei comodi requester di Arp, si può notare come questo venga eseguito con una certa... intelligenza. Chi abbia già adoperato un cosiddetto «monitor» di linguaggio macchina, sa come il codice disassemblato venga normalmente esibito assolutamente «nudo», con i riferimenti ad altre locazioni del programma espressi da valori assoluti o, nella migliore delle ipotesi, mediante rappresentazione degli «scostamenti» (offset) rispetto all'indirizzo di inizio del programma. «Resource», senza ancora alcun intervento da parte dell'utente, propone già qualcosa di più, riconoscendo la presenza di particolari messaggi definiti come costanti, come potrebbe essere la descrizione del nome di una libreria. Assumendo come esempio una banale sezione di file, dopo il suo caricamento si avrebbe insomma una visualizzazione di questo tipo:

```
MOVE.L #doslibrary.MSG,A1
MOVEQ #0,D0
MOVE.L 4,A6
JSR −$228(A6)
TST.L D0
BEQ START+$40
.... omissis ....
RTS
doslibrary.MSG dc.b 'dos.library',0,0
```

L'etichetta che identifica la posizione della stringa «DosLibrary» sarà già stata definita da «Resource», e così anche ogni eventuale riferimento ad essa in qualunque punto del listato. Per rendere veramente chiaro e leggibile il disassemblato è però necessario qualche altro passaggio. Ancora in totale automatismo, è per esempio possibile eliminare uno scomodo riferimento come quello presente nell'istruzione Beq Start+$40. Volendo, si potrebbe sfruttare l'opzione Show Offset del menu O1 (= Options 1), che visualizza a sinistra dei codici la loro posizione relativa all'inizio del programma. In tal modo ci si renderebbe subito conto del punto di arrivo del salto condizionato Beq, ma, anche in questo caso, l'uso di una Label risulta l'optimum. Niente di più facile: si seleziona Disassemble dal menu Project, e si vedrà apparire (per esempio) a sinistra di RTS una etichetta siglata lbc000040, mentre anche l'operando di Beq sarà diventato lbc000040.

Da questo momento, ha inizio il lavoro più «certosino», nel quale occorre anche impegnare in prima persona le proprie cognizioni, ma sempre con un grosso aiuto da parte di «Resource». Intanto, si potrebbe cominciare con l'assegnare all'etichetta appena creata un nome dal significato più intuitivo; essendo associata ad un Rts, questo potrebbe per esempio essere «uscita». In pratica, bisognerà prima di tutto far scrollare lo schermo in modo che l'istruzione Rts (e relativa label) diventi la prima linea dello schermo, operazione da effettuare in tutti quei casi in cui occorre agire su di una singola riga: la prima dello schermo sarà quella «attiva», anche se la cosa non viene evidenziata. Ora, selezionando Edit Single/Label dal menu L (= Labels), si potrà digitare nel requester proposto da «Resource» il nuovo nome. Anche in questo caso, verranno aggiornati automaticamente tutti i riferimenti alla label presenti in qualunque altro punto del disassemblato, a velocità stratosferica. A questo proposito va detto che «Resource», proprio per privilegiare la velocità, impegna una notevole quantità di memoria, in rapporto ovviamente alle dimensioni del file da trattare. 1 Megabyte di ram è da ritenersi il minimo indispensabile, e per programmi di 30-40 Kb occorre considerare una disponibilità di circa 2 Megabyte.

Tornando ancora all'esempio proposto prima, non può non essere citata la più importante delle caratteristiche di «Resource», ovvero l'identificazione quasi del tutto autonoma delle funzioni di libreria richiamate dal programma in esame.

Prendiamo in considerazione l'istruzione Jsr −$228(A6): il fatto che sia preceduta da un Move.l 4,A6 lascia intuire che è coinvolta la libreria Exec, ma quale funzione? Nessun problema: si «porta» l'istruzione in cima allo schermo con i tasti cursore, quindi si seleziona il menu S2 (Symbols2), scegliendo dal lungo elenco (ordinato alfabeticamente) che appare posizionandosi con il mouse alla voce E-G, l'opzione Exec Library. L'istruzione di salto sarà diventata Jsr _LvoOpenLibrary(A6). Se il file contenesse subito dopo un nuovo salto ad una funzione di libreria con il registro A6 contenente l'output della precedente chiamata, sapremmo che stavolta riguarderebbe la Dos Library appena aperta, e quindi si sceglierebbe dal menu S2 l'opzione D/Doslibrary... e così via fino al chiarimento di tutti i punti oscuri del disassemblato. Salvando poi il file in formato sorgente, si potrà decidere (menu SV/Symbol Table) se farvi inserire o meno una serie di EQU (o anche di XREF) riferita a tutte le funzioni adoperate.

Fin qui non si è accennato che ad una minima parte delle innumerevoli implementazioni di «Resource», che vanno da una completa ridefinizione dei cosiddetti «hot keys», a svariate funzioni di editing sul disassemblato o, ancora, alla creazione (automatica!) di Macro in grado di rieffettuare tutte le operazioni compiute nel trattamento di un file.

GLEN MCDIARMID
28 Marginson Street
Ipswich, QLD
Australia 5405

http://www.oldgamesitalia.net/
http://www.oldgamesitalia.net/

http://www.oldgamesitalia.net/

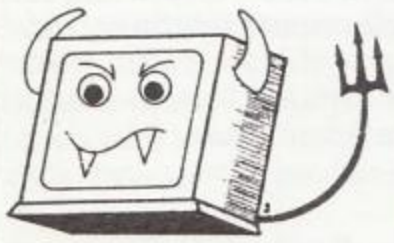
LE TENTAZIONI DI AMIGA
solo per adulti

■ AMI PORNO SHOCK
Due dischetti con le immagini più hard mai viste sul tuo computer e un'animazione che metterà a dura prova il tuo joystick!

Lire 25mila

■ PORNO FILM
È il conosciutissimo (per chi ce l'ha...) AmigaByte PD7: un dischetto eccezionale con tre film. Julie, Bridget e Stacy i tre titoli. I primi due di animazioni, il terzo un favoloso slideshow con definizione e dettagli che stupiscono.

Lire 10mila



Per ricevere AmiPornoShock oppure PornoFilm basta inviare vaglia postale ordinario ad AmigaByte, C.so Vitt. Emanuele 15, Milano 20122. Specifica sul vaglia stesso la tua richiesta (Shock oppure Film) e naturalmente il tuo indirizzo. Per un recapito più rapido aggiungi lire 3mila e chiedi spedizione espresso!

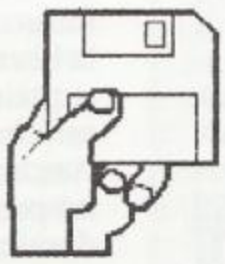
nuovissimo CATALOGO SOFTWARE PUBBLICO DOMINIO

★ *Il catalogo viene continuamente aggiornato con i nuovi arrivi!!!*

CENTINAIA DI PROGRAMMI

UTILITY
GIOCHI
LINGUAGGI
GRAFICA
COMUNICAZIONE
MUSICA

IL MEGLIO DEL PD
e in più
LIBRERIA COMPLETA
FISH DISK 1 - 460



★ SU DISCO ★

Per ricevere il catalogo su disco invia vaglia postale ordinario di lire 10.000 a AmigaByte C.so Vitt. Emanuele 15 20122 Milano

PER UN RECAPITO PIÙ RAPIDO aggiungi L. 3.000 e richiedi SPEDIZIONE ESPRESSO

Sì,
anche tu puoi collaborare ad Amiga Byte!

Con articoli, megagame, idee...

★

La redazione è a tua disposizione per vagliare ogni lavoro

★

Invia una scaletta di quello che pensi di poter fare o un dischetto con le tue creazioni

★

Spedisci ad AMIGABYTE c.so Vitt. Emanuele 15 20122 Milano

http://www.oldgamesitalia.net/
http://www.oldgamesitalia.net/

http://www.oldgamesitalia.net/

Le funzioni della libreria ARP

Tutti i vantaggi del pacchetto Arp e della sua pratica libreria, facilmente installabili sia dai programmatori più esperti che da quelli alle prime armi.

di GIUSEPPE SACCHI

Com'è ben noto, l'acronimo **Dos** indica generalmente il sistema operativo relativo al disco (**Disk Operating System**). Su Amiga, tuttavia, il termine Dos ha un duplice significato; indica infatti, a livello più basso, una particolare libreria di funzioni, che facilita gli accessi al disco e gestisce i *file* e le *directory* in modo gerarchico (struttura ad albero); nel contempo, a livello più

alto, fa riferimento ai comandi utilizzabili tramite una delle due interfacce tipiche di Amiga, vale a dire il **Cli** (o la più moderna **Shell**), che ricorda vagamente l'interfaccia MsDos: questi comandi, che altro non sono che piccoli programmi, sono contenuti in una apposita directory identificata come un *dispositivo logico* (c:). Come illustriamo a parte, la «**dos.library**» occupa un posto cruciale nell'ambiente operativo Amiga, anche se le routine contenute in **Exec** (la libreria che tiene sotto controllo tutto il sistema operativo) ci possono consentire, volendo, di scrivere programmi che facciano a meno della libreria Dos e, ancor più, dei cosiddetti comandi-Dos.

LE CARENZE DEL DOS

Qualcuno (soprattutto creatori di giochi) è stato spinto ad operare in tal senso, sconcertato dalle carenze del Dos di Amiga; altri hanno addirittura coniato epiteti poco gradevoli, come «Amiga dog»; altri ancora (non molti, per fortuna), spinti da particolari esigenze, si sono sostituiti *in toto* al sistema operativo.

Personalmente non crediamo affatto che il Dos di Amiga sia da buttare, tuttavia bisogna ammettere che alcune carenze esistono. In primo luogo, la libreria e i comandi Dos sono stati scritti in **BCPL**, un linguaggio che ha fatto ormai il suo tempo e che contrasta nettamente con il resto del sistema, rendendo più complicate e difficili le comunicazioni con le altre librerie e con i programmi-utente, solitamente scritti in C o in assembler: inoltre, la libreria del Dos non è particolarmente ricca di funzioni e qualcuna di esse agisce in modo contorto e poco corretto (come **Execute()**, che sarebbe preferibile non usare). Anche i comandi-Dos hanno i loro bravi difetti: occupano parecchio spazio su disco, non sono velocissimi, rendono alcune operazioni particolarmente ripetitive, etc.

I COMANDI ARP

A partire dal 1987 un gruppo di volonterosi programmatori coordinati dall'abile Charlie Heath, della Microsmith Inc., ha messo mano ad un progetto per rimpiazzare il Dos dell'Amiga; questo progetto ha preso appunto il nome di **ARP (AmigaDos Replacement Project).**

«Arp» è costituito essenzialmente da una libreria (**Arp. Library**) che incorpora tutte le funzioni della «dos.library», alle quali sono state aggiunte numerose altre funzioni che cercano di colmare le lacune precedenti. Con l'ausilio di questa libreria e con l'uso del linguaggio più efficiente in assoluto, l'*assembly*, è poi stata riscritta la stragrande maggioranza dei comandi DOS, quelli presenti nella directory «c:», come *copy, dir, delete*, e così via. Con una iniziativa estremamente lodevole, si è infine reso «Arp» di pubblico dominio.

Attualmente, l'ultima versione del pacchetto «Arp» in circolazione è la 1.3, mentre la libreria «Arp.Library» è giunta alla versione 39.1. «Arp» è oggi particolarmente

http://www.oldgamesitalia.net/ http://www.oldgamesitalia.net/

http://www.oldgamesitalia.net/

efficiente ed apprezzato in tutto il mondo da numerosi programmatori ed utenti; inoltre non ha messo in evidenza particolari *bug* o problemi, e la sua utilità e praticità lo rendono quasi indispensabile sui nostri dischetti. Sono sempre più numerosi i programmi che hanno bisogno della presenza della «arp.library» per poter girare, in particolare fra il *software* di pubblico dominio; anche alcuni validissimi prodotti commerciali, come i linguaggi della HiSoft, sono ricorsi all'ausilio di questa potente libreria, con paricolare riferimento al suo pratico selettore di file (**file requester**).

INSTALLIAMO ARP

Il pacchetto «Arp» viene distribuito tramite le collezioni più note e complete di software di pubblico dominio per Amiga: chi ne fosse sprovvisto, può richiederlo anche presso la nostra redazione inviando un vaglia postale di 10 mila lire.

È anche possibile ottenere tutti i file relativi ad «Arp» acquistando l'*editor* di testo «**TxEd**», realizzato dallo stesso Charlie Heath e distribuito dalla Micro-Smiths.

Un ambito in cui sono particolarmente rilevanti le carenze di AmigaDos è quello degli *editor* di testo; nella *directory* «c:» del disco WorkBench, infatti, sono contenuti due editor («Ed» ed «Edit») macchinosi e poco pratici, completamente privi di una interfaccia intuitiva (cosa ci stanno a fare su di un computer come Amiga?). «TxEd», pur essendo di piccole dimensioni, supera brillantemente questi limiti: varrebbe la pena di installare la libreria «Arp» solo per poter utilizzare questo programma!

Installare «Arp» in modo personalizzato risulta facile per tutti attraverso l'uso di un apposito programma che permette di scegliere, con il solo ausilio del mouse, quali file copiare sul disco di sistema. Operando su di una copia del WorkBench, è caldamente consigliabile la sostituzione di tutti i comandi Dos originali, ed inoltre l'installazione della particolare «shell» fornita da «Arp», denominata «**AShell**», che si dimostra particolarmente efficiente. A lavoro ultimato, nonostante la presenza in più, nella directory «libs:», del file «arp.library», noterete di aver guadagnato oltre 47 KByte di spazio sul dischetto WorkBench ed il vostro ambiente di lavoro sarà sensibilmente migliorato sotto il profilo della velocità e dell'efficienza.

MAGGIORE POTENZA

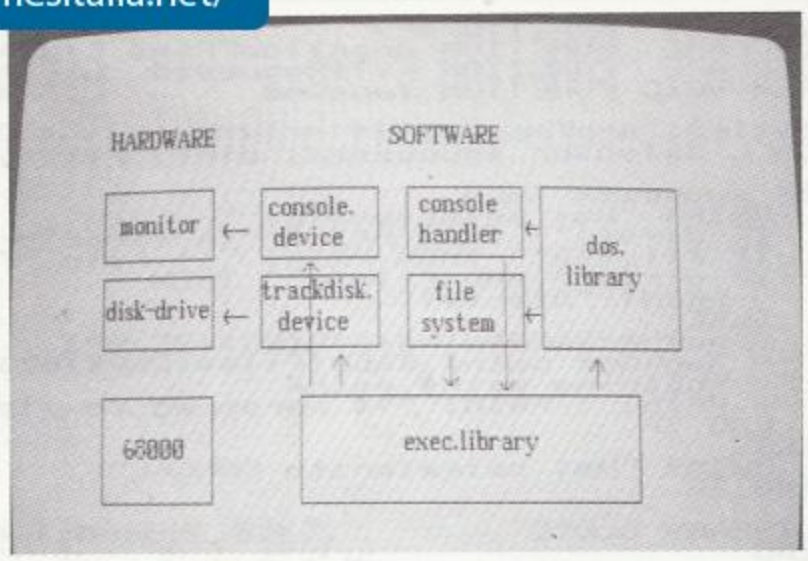
I comandi-DOS forniti da «Arp» non solo sono molto più compatti di quelli «ufficiali» della *Commodore*, ma risultano anche più potenti. Molti comandi supportano ora argomenti multipli; solo per fare un esempio, se per effettuare due assegnazioni prima dovevamo scrivere:

1> assign Mario: df0:
1> assign Beppe: ram:

ora possiamo scrivere:

1> assign Mario: df0: Beppe: ram:

Oltre che risparmiarci fatica, questa possibilità può



LA DOS.LIBRARY ED IL SISTEMA

Lo schema qui illustrato cerca di raffigurare in modo drasticamente semplificato i rapporti che la dos.library intrattiene con il resto del sistema e con l'*hardware*. A sinistra abbiamo rappresentato l'*hardware*, al centro e a destra il *software*. In basso abbiamo rispettivamente la CPU (il 68000) ed il «cuore» del *software* (la libreria Exec); più sopra abbiamo collocato la libreria Dos, il *device-handler* (il Filing System), e la libreria-device *trackdisk.device*. Exec si serve del *trackdisk.device* per scrivere e leggere da disco: è il procedimento usato, per esempio, dal comando *format*; se vogliamo usare il Filing System (sistema di archiviazione) standard, sarà però preferibile aprire (sempre attraverso Exec) la libreria Dos. Questa libreria (quando utilizzeremo *Open* o *Read*, etc.) comanderà per noi Exec, il quale spedirà poi un messaggio al processo creato dal *trackdisk.device* per gestire il drive da noi scelto; nel fare questo il DOS utilizzerà una sorta di filtro che si interpone tra esso ed Exec e che prende il nome di «gestore del dispositivo» (*device-handler*) e che, nel caso in cui il dispositivo sia il drive, non è altro che il File-System. Questo sistema, che può sembrare eccessivamente complicato, permette però di gestire come se fossero semplici file (con OPEN, CLOSE, etc.) le più diverse periferiche: basterà scrivere l'appropriata libreria-device (da collocare nella directory DEVS) ed il relativo *device-handler* (che va nella directory L); nello schema abbiamo fatto anche riferimento alla console del CLI, ma si potrebbe agire in questo senso anche nei confronti di un'altra periferica, o per gestire in modo diverso una periferica già esistente. Per esempio, il notevole Messy-Dos, pubblicato da AmigaByte sul dischetto allegato al fascicolo 23, opera attraverso una libreria-device (*Messydisk.device*) ed un *device-handler* (*MessyFileSystem*) che, copiati nelle directory giuste e resi attivi tramite il comando *Messymount*, permettono al Dos di Amiga di leggere un disco in formato MsDos/Ibm senza il minimo problema. Per precisione, si tenga comunque presente che lo schema qui presentato è piuttosto riduttivo e che il rapporto tra le librerie-dispositivo e l'*hardware* non è in realtà così diretto come potrebbe sembrare.

I comandi Arp che si sostituiscono a quelli originali di AmigaDos, mantenendo la stessa sintassi.

http://www.oldgamesitalia.net/
http://www.oldgamesitalia.net/

http://www.oldgamesitalia.net/

FileRequest.Bas

```
LIBRARY "arp.library"
DECLARE  FUNCTION ArpAllocFReq& LIBRARY    'Alloca struttura FileRequest
DECLARE  FUNCTION FileRequest&  LIBRARY    'lancia il FileRequest
REM  VOID FUNCTION TackOn&          LIBRARY    'unisce directory+file

title$="ArpFromBasic!"
CALL Selector (success%, dir$, file$, title$)

IF success% THEN
   PRINT "Hai selezionato: "
   IF dir$<>"" THEN PRINT "la directory "dir$
   IF file$<>"" THEN PRINT "il file "file$
   IF dir$="" AND file$="" THEN
     PRINT "niente!"
   ELSE
     path&=SADD(dir$+STRING$(32,0))
     TackOn& path&,SADD(file$+CHR$(0))
     IoString path$,path&
     PRINT : PRINT "il percorso completo è: "path$
   END IF
ELSE
   PRINT "Hai selezionato CANCEL"
END IF

LIBRARY CLOSE
END

REM Subroutine che richiama il FILE REQUESTER.
REM xtitle$ contiene il titolo-requester; xdir$ restituisce nome directory;
REM xfile$ restituisce nome file; in rc% valore booleano (vero/falso)

SUB Selector (rc%,xdir$,xfile$,xtitle$) STATIC
rc%=0
REM SE SI USA AC-BASIC AGGIUNGERE () ALLA FINE DELLA LINEA SEGUENTE
IF freq&=0 THEN freq&=ArpAllocFReq&
IF freq&=0 THEN EXIT SUB
POKEL freq&,SADD(xtitle$+CHR$(0))

s&=FileRequest&(freq&)

IF s&<>0 THEN
   rc%=-1
   p&=PEEKL(freq&+8)
   IoString xdir$,p&
   IoString xfile$,s&
END IF
END SUB

REM Convertitore di stringa da C

SUB IoString (s$,p&)  STATIC
c$="" : s$="" : i%=0

WHILE c$<>CHR$(0)
   s$=s$+c$
   c$=CHR$(PEEK(p&+i%))
   i%=i%+1
WEND

END SUB
```

Una dimostrazione della grande versatilità di Arp: il file requester richiamato in due programmi scritti in Basic e C, (una versione in Assembler è inoltre sul disco allegato a questo fascicolo). Esempi analoghi possono essere realizzati in Modula 2 e in svariati altri linguaggi.

velocizzare ad esempio l'esecuzione di sequenze di avviamento (*startup-sequence*) molto lunghe, permettendo di caricare i comandi da disco una sola volta.

Una nota dolente di AmigaDos è costituita dalla complicata e poco pratica gestione dei **pattern** di ricerca per i nomi di file e directory, limitata oltretutto ad alcuni comandi e preclusa ad altri (ad esempio "Type"). Soprattutto chi proviene dal mondo MsDos si scontra con i caratteri speciali %, ?, #, e rimpiange spesso la **wildcard** costituita dall'asterisco (*) che permette, nei PC IBM e compatibili, di indicare una qualsiasi sequenza di caratteri. I comandi «Arp» rappresentano, soprattutto in questo senso, un enorme passo in avanti, perché permettono di usare sia i caratteri speciali tradizionali Amiga DOS sia l'asterisco; inoltre, i caratteri speciali sono utilizzabili praticamente con tutti i comandi, al contrario del DOS standard. Diremmo che a questo modo i comandi «Arp» non solo risultano migliori di quelli Amiga DOS *Commodore*, ma anche di quelli MS-DOS; se avete «Arp» installato e date il seguente comando:

1> Dir *file.*

avrete una lista dei file, il cui nome comincia con una sequenza qualsiasi di caratteri e termina con la stringa «file» e che sono dotati di una qualsiasi estensione. Se da MS-DOS scrivete:

A> Dir *file.*

il comando sarà interpretato poco correttamente come **Dir *.*** e verrà visualizzata una lista di tutti i file.

Due comandi molto usati, e che sotto Amiga DOS standard quasi assurdamente non riconoscevano le *wildcards*, sono «**Type**» e «**Rename**». Immaginate di avere, nella directory-radice del vostro disco, una decina di file di immagini IFF con l'estensione «.iff» e di voler mettere un po' d'ordine trasferendoli tutti nella directory «immagini»; con il DOS tradizionale dovreste usare dieci (!) volte il comando «Rename», una volta per ogni file; con «Arp» basterà scrivere:

1> Rename *.iff immagini/*.iff

Una caratteristica gradevole di AmigaDos consiste nella possibilità di chiedere la corretta sintassi di un comando ricorrendo all'argomento «**?**». «Arp» mantiene e migliora questa caratteristica, poiché in caso di errore stampa messaggi più completi ed espliciti.

Come si vede, lavorare con «Arp» può essere piuttosto piacevole e può aiutarci in molte circostanze. Anche i comandi aggiuntivi possono tornare utili. Di «Ashell» abbiamo già parlato; «**Arun**» si dimostra più elastico ed efficiente del tradizionale «Run»; inoltre tramite «**Ares**» c'è la possibilità di rendere residenti i comandi DOS, con uno stretto controllo sul loro funzionamento, superiore a quello effettuato dal comando standard «Resident»; peccato solo che questo controllo sia talmente stretto da creare serie difficoltà con i programmi non-Arp.

ARP PER I PROGRAMMATORI

Sia come utenti finali che come programmatori possiamo trarre giovamento da «Arp» e dalla sua libreria di funzioni. Come programmatori non possiamo fare a meno di considerare che «Arp», oltre che le normali funzioni DOS, ci mette a disposizione circa settanta funzioni nuove, in genere molto utili, grazie alle quali possiamo alleggerire notevolmente il lavoro e rendere più piccolo il nostro codice: inoltre possiamo comunicare alla libreria

http://www.oldgamesitalia.net/
http://www.oldgamesitalia.net/

http://www.oldgamesitalia.net/

FileRequest.C

```
/* Per Lattice C, facilmente adattabile ad Aztec:
   compilare cosi:

   lc -v FileRequest

   BLink lib:arpc.o FileRequest.o LIB lib:lc.lib lib:arp.lib nd

*/

#include <exec/types.h>
#include <libraries/arpbase.h>
#include <proto/exec.h>
#include <proto/dos.h>
#include <proto/arp.h>

struct ArpBase *ArpBase;

main()
{
struct FileRequester *freq;
char                 *file;
char                 *path;

/* tentiamo di aprire ARP, versione 39 */
ArpBase=(struct ArpBase *)OpenLibrary("arp.library",39L);
if (!ArpBase)
   {
   Write(Output(),"necessaria \"arp.library\" ver. 39 o superiore!\n",51L);
   exit(5);
   }

/* allochiamo memoria struttura FileRequester e stringa con tracciamento
   risorse (ci penserà ARP a liberare la memoria) */

if ((freq=(struct FileRequester *)ArpAllocFreq())
                        && (path=(char *)ArpAlloc(256)))
   {
   freq->fr_Hail="ArpFromC!";                    /* titolo requester          *
/
   if (file = (char *)FileRequest(freq))         /* passiamo il controllo
                                                    al requester *
/
      {
      strcpy(path,freq->fr_Dir);
      TackOn(path,file);                         /* colleghiamo dir + file    *
/
      if (path[0]=='\0') strcpy(path,"NIENTE");
      Printf("Hai selezionato: %s.\n",path);     /* printf di ARP (iniziale
                                                    maiuscola) *
/
      }
   else
      Puts("Hai selezionato CANCEL");            /* puts di ARP (come sopra)  *
/
   }
else
   Puts("Memoria insufficiente!");

CloseLibrary((struct Library *)ArpBase);
}
```

di tenere traccia di tutte le risorse che allochiamo, in modo che le possa poi «scaricare» automaticamente; naturalmente «Arp» non fa miracoli e, se il nostro codice è bacato, tale rimarrà anche se utilizzeremo questa libreria, a meno che il *bug* non si trovi in un modulo che viene rimpiazzato da una funzione di «Arp».

Tenete però presente che l'utilizzo della «arp.library» non comporta solo vantaggi: su disco il vostro codice risulterà più piccolo, ma in esecuzione il programma avrà probabilmente bisogno di maggior memoria (se poi la libreria non è già stata caricata da qualche altro *task*, all'atto della sua apertura verranno occupati esattamente 18 Kbyte).

Considerate anche che il difetto maggiore di questa libreria (non dipendente dai suoi autori) è di non trovarsi in ROM ma su disco, per cui qualche utente sprovveduto può copiare il file eseguibile ma non la libreria, trovandosi poi in difficoltà al momento di lanciare il programma.

Inoltre (ultimo argomento ma di non scarso rilievo) l'uso di questa libreria presuppone un certo spirito di adattamento: se il selettore di file è troppo piccolo o troppo grande, se non ci piacciono le date sempre e comunque in inglese, o se qualche altra funzione non ci va a genio, c'è ben poco da fare, dobbiamo sacrificare le nostre esigenze sull'altare della fatica risparmiata e delle dimensioni del codice eseguibile. Diciamo che la scelta di utilizzare o non la «arp.library» dipende dal programmatore e dalle sue esigenze, e che un programma può essere ugualmente valido sia che la usi o no, purché la scelta sia stata operata con coerenza.

Poiché la libreria è una normale **runtime library** di Amiga e ne segue tutti i dettami, possiamo chiamare le sue funzioni utilizzando qualsiasi linguaggio; se in qualche caso, come nel Basic, gli autori di «Arp» non ci offrono il necessario file di interfaccia, basta ricorrere al classico file «**.fd**» per costruircene uno. Il linguaggio di elezione per utilizzare questa libreria è comunque il C, per il quale esistono numerosi file (file «include», file «oggetto», librerie per il *linking* col nostro codice oggetto); con rammarico dobbiamo dire che i file adatti al compilatore della Lattice (ora più noto come Sas C), per quanto siano presenti, ci sono sembrati meno completi rispetto a quelli dedicati al compilatore Aztec della Manx. Ciò deriva probabilmente dal fatto che il *software* è stato sviluppato con il prodotto della Manx, ma comunque il fatto può essere fastidioso per chi utilizzi il Lattice. Anche per l'assembler non ci sono problemi, essendo fornito un ottimo e completo file «include». Per sapere come utilizzare la libreria ricordiamo che i file ASCII sono tutti ampiamente commentati. Sul disco con cui viene distribuito «Arp» esistono anche due dettagliati manuali, uno per l'utente che voglia far uso dei comandi sostitutivi ed uno per il programmatore; in quest'ultimo manuale (costituito da diversi file di testo) viene esaminata accuratamente ogni funzione della libreria, con la specifica dei registri usati e delle modalità di utilizzo; si sente comunque la mancanza di qualche programma di esempio.

LE FUNZIONI PRINCIPALI

L'attrezzo forse più apprezzato fornito da «Arp» è il **File Requester**, un selettore di file molto comodo e potente, utilizzabile con una semplice chiamata di funzione; visto l'interesse rivestito da questo argomento, abbiamo inserito sul dischetto allegato a questo fascicolo alcuni semplici esempi su come utilizzarlo da C, da Basic e da

http://www.oldgamesitalia.net/
http://www.oldgamesitalia.net/

http://www.oldgamesitalia.net/

Assembler. Alla funzione *FileRequest* va passato solo l'indirizzo di una struttura apposita, articolata come segue:

```
struct FileRequester {
   BYTE *fr_Hail; /* titolo requester */
   BYTE *fr_File; /* stringa con nome del file */
   BYTE *fr_Dir; /* stringa con nome directory*/
struct Window *fr_Window; /* finestra di rifer.
   o ZERO */
   UBYTE fr_FuncFlags; /* flags */
   UBYTE fr_Flags2; /* altri flags... */
   VOID (*fr_Function)(); /* nostra funzione */
   WORD fr_LeftEdge; /* angolo sinistra */
   WORD fr_TopEdge; /* angolo in alto */
   };
```

Nell'ultima versione della libreria è stata aggiunta la funzione **ArpAllocFReq()**, che alloca automaticamente una struttura di questo tipo, fornendo anche i byte necessari per le stringhe destinate a contenere i nomi di file e directory; poichè «Arp» tiene traccia di queste allocazioni, non dovremo mai preoccuparci di liberare questa memoria.

Nei casi più semplici basterà inizializzare il primo membro della struttura (il titolo), che potrà poi essere passata a *FileRequest*; se il membro *fr_Window* contiene l'indirizzo di una particolare finestra, il *requester* verrà agganciato ad essa, potendo così comparire anche in schermi che non siano quello del Workbench; volendo, possiamo anche inserire nel membro *fr_Function* l'indirizzo di una funzione tutta nostra (scritta preferibilmente in C), che venga poi utilizzata all'interno del requester; con queste tecniche possiamo ottenere particolari prestazioni (per esempio rendere «invisibili» al selettore tutti quei file che non hanno una certa estensione, oppure aggiungere dei *gadget*).

Subito dopo la *FileRequest*, in ordine di gradimento, si può porre la funzione **GADS()**, che si occupa di prelevare la linea di comando battuta da Cli e di esaminare se è conforme alla sintassi CLI (*template*) che noi stessi abbiamo stabilito. *GADS()* va usata in questo modo:

num_arg = GADS(linea,lung,aiuto,argv,sintassi)

dove «num_arg» è il numero di argomenti riscontrato dalla funzione; «linea» è il puntatore alla stringa contenente la linea di comando; «lung» è la lunghezza di tale linea; «aiuto» è una stringa di aiuto che la funzione si occupa di visualizzare se l'utente è in difficoltà; «argv» è il vettore di stringhe che punterà agli argomenti così come sono stati scanditi dalla funzione; «sintassi» è una stringa che contiene la descrizione della sintassi secondo lo standard Amiga. Ricordiamo per inciso che con «/A» si indicano gli argomenti che sono sempre presenti, con «/K» gli argomenti che richiedono una parola-chiave ed un parametro, con «/S» quelli che richiedono solo una parola-chiave. La funzione *GADS()* svolge un compito altrimenti estremamente gravoso, e ci permette di analizzare facilmente e con precisione gli argomenti digitati dall'utente sulla linea di comando del CLI.

Va ricordato, per chi programma in C, che i codici di *startup* elaborati dagli autori di «Arp» chiamano autonomamente tale funzione per cui, se ci serviamo di essi, basterà inizializzare le stringhe di sintassi e di aiuto per il nostro programma, senza invocare direttamente *GADS()*; con l'assembler, i valori «linea» e «lung» si troveranno automaticamente nei registri a0 e d0 al momento del lancio del programma; con il Basic, le cose sono praticabili con facilità solo se si usa il compilatore Hisoft.

MEMORIA E MULTITASKING

Si è anche accennato alla possibilità che «Arp» offre di gestire per conto nostro le risorse (e la relativa memoria) che abbiamo utilizzato nel nostro programma. Premesso che è una pessima abitudine «dimenticarsi» di rilasciare memoria, di chiudere *file* aperti in precedenza, o cose del genere, attraverso «Arp» c'è effettivamente la possibilità di incaricare la libreria di svolgere questo noioso compito. Tutte le funzioni inizianti con "Arp" come, per esempio, *ArpAllocMem()*, *ArpLock()*, *ArpOpen()*, comportano che la libreria tenga traccia delle risorse in una lista appropriata, e che queste vengano rilasciate nel momento in cui il nostro programma effettua una chiamata alla funzione *ArpExit()*, per terminare l'esecuzione, oppure chiuda la libreria «Arp» con un semplice *CloseLibrary()*.

Considerata la notevole mole di codice che funzioni come *printf()* generano in linguaggio C, «Arp» ci fornisce alcuni pratici e veloci sostituti, come *Printf()*, *FPrintf()*, *SPrintf()*, *Puts()*, *Strcmp()*, *ToUpper()*, che, oltre che generare pochissimo codice, funzionano (quasi) altrettanto bene e sono così disponibili anche fuori dall'ambito del C.

Si è anche osservato in precedenza che la libreria DOS risulta piuttosto «scarna» e non molto ricca di funzioni. Anche in questo senso «Arp» ci viene incontro; per esempio, *InitStdPacket()* e *SendPacket()* ci permettono di strutturare e «spedire» i *pacchetti* del DOS; *SetEnv()* e *GetEnv()* fissano e prelevano variabili di ambiente, solo da poco introdotte nel sistema di Amiga; particolarmente utile può dimostrarsi la funzione *Assign()*, che ha una sintassi praticamente uguale a quella del relativo comando DOS e attraverso la quale possiamo anche sapere se l'assegnazione ha funzionato. È presente anche tutta una serie di *routine*, tra le quali spiccano *ASyncRun()* e *SyncRun()* che permettono di lanciare un programma in modo asincrono (multitasking) ed in modo sincrono; chi volesse utilizzare *ASyncRun()* si ricordi comunque di inizializzare una particolare struttura (*Process Control Block*), che in questa sede risulta troppo complessa da illustrare ma che è comunque ben documentata nel manuale.

LE FUNZIONI DI MANIPOLAZIONE

Un po' meno entusiasti lasciano le funzioni di manipolazione delle date, non perché abbiano *bug* o cose del genere, ma per il principio sul quale sono basate; attraverso *StampToString()* si converte la struttura DOS *DateStamp* in una stringa, attraverso *StringToStamp()*, si opera in senso contrario; una particolare opzione permette di variare a piacimento il formato delle date e, a detta degli autori di «Arp», di renderle «internazionali»; peccato però che le stringhe siano comunque in inglese: così la tanto vantata internazionalità va a farsi benedire.

Naturalmente le possibilità di «Arp» non si fermano qui: possiamo prelevare con relativa facilità liste di dispositivi o di file/directory, ordinare velocemente qualsiasi genere di oggetti (numeri o stringhe), convertire stringhe dal formato BCPL a quello del C e viceversa, verificare se una certa stringa corrisponde ad una certa *maschera* o *pattern* (indicata dall'utente anche con l'uso dell'asterisco), e via dicendo.

Nel complesso, se si esclude qualche limite più che plausibile (non ultimo il fatto di risultare pressoché inutile per chi lavora con il nuovo sistema operativo 2.0), «Arp» si dimostra uno strumento utilissimo, anzi spesso indispensabile; il tempo perso per conoscerlo più a fondo sarà senz'altro ben ripagato, sia per l'utente «normale» che per il programmatore.

http://www.oldgamesitalia.net/
http://www.oldgamesitalia.net/

http://www.oldgamesitalia.net/

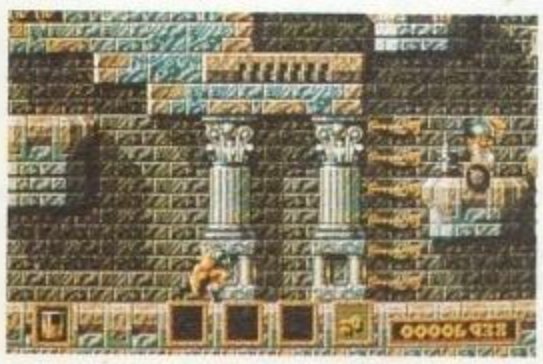
Software Express

di Marco Brovelli

GODS

Un nuovo genere di gioco nato recentemente sull'onda di programmi come «Popolous» è quello dei «simulatori divini», ovvero dei giochi che mettono il giocatore nella posizione di un'entità onnipotente con il compito di decidere le sorti di interi pianeti o popolazioni. I più recenti esponenti di questa categoria sono «Sim City», «PowerMonger» e «Lemmings». Tuttavia, a dispetto del nome, «Gods» non appartiene a questo genere di videogioco. Si tratta infatti di un arcade adventure, con l'enfasi posta sul versante arcade, nel quale gli dei del titolo sono quelli dell'Olimpo dell'antica Grecia.
Opera del team di programmatori Bitmap

Brothers, celebre per giochi arcade come «Xenon II», «Gods» è un capolavoro sotto il profilo estetico: gli sprite, gli sfondi e le animazioni sono di primissima qualità, e la giocabilità non è da meno. Pur non essendo tremendamente originale come meccanismo di gioco, «Gods» si distacca dalla tradizionale formula dello «stermina tutti i nemici e completa il livello annientando il super-mostro finale» grazie alla presenza di alcuni elementi di strategia. Oltre che lottare contro arpie, gorgoni, centauri ed altre orrende creature mitologiche, il muscoloso eroe di «Gods» deve raccogliere oggetti, attivare interruttori, evitare trappole, teletrasportarsi qua e là per i livelli ed usare una buona dose di ragionamento e di memoria prima di gettarsi ciecamente nella



mischia.
«Gods» ricorda molto «Turrican» come giocabilità e tipo di azione, con un'ambientazione mitologica e l'aggiunta di elementi strategici (negozi nei quali acquistare armamenti più potenti, problemi da risolvere, etc.). Un prodotto sicuramente riuscito ed allettante.

HUNTER

Una volta tanto, la grafica vettoriale non viene usata per un simulatore di volo o per un gioco di combattimento tra carri armati, bensì per un arcade-adventure (Activision) di ambientazione militare imparentato più con «Castle Master» che con «Battle Command». A differenza dei giochi della serie Freescape della Incentive, come «Total Eclipse», l'azione in «Hunter» non è ripresa dal punto di vista soggettivo del protagonista ma dall'esterno, con possibilità di zoomare per avvicinare o allontanare l'inquadratura.
Il personaggio manovrato dal giocatore in questo gioco Activision è un soldato fatto di solidi vettoriali che, a giudicare dalla camminata, sembra essere imparentato con l'Arnold Schwarzenegger di «Terminator». Nonostante l'apparente rigor mortis che lega i suoi movimenti, il nostro eroe è in grado di compiere un'impressionante varietà di azioni, che variano dal guidare un'automobile al pilotare una barca, senza

contare il nuotare in mare e persino l'andare in windsurf.
All'inizio di ogni partita, viene assegnata una missione da compiere entro un tempo limite. Sia che si tratti di addentrarsi in territorio nemico per distruggere un deposito di carburante o di impadronirsi di un veicolo, occorrono pazienza ed intuito per raggiungere l'obiettivo e tornare sani e salvi al quartier generale. Oltre che un log book, che funge da mappa, potete portare con voi una notevole quantità di oggetti ed armi, o raccogliere altro lungo la strada.
Le mosse del personaggio sono comandate dal joystick, mentre le altre funzioni vengono

impartite tramite il mouse e la tastiera. Nonostante la quantità di comandi disponibili, questo metodo di controllo non risulta affatto scomodo.
La grafica è piuttosto rozza ma sufficientemente veloce, mentre il suono è pressoché inesistente. In virtù comunque della varietà di missioni e dell'originalità del tema e del meccanismo di gioco, «Hunter» si guadagna indubbiamente una promozione sul campo.

REVELATION

Adoratori dei puzzle con migliaia di pezzi, fanatici del cubo di Rubik, appassionati dei rompicapo di qualsiasi genere e difficoltà, ecco il gioco che fa per voi: «Revelation», distribuito dalla Krisalis, appartiene a quella categoria di videogame che, una volta tanto, fa venire l'irrefrenabile desiderio di fracassare il monitor a colpi di joystick a causa della sua diabolica complessità e non per lo scarso livello della grafica o della giocabilità. Come in tutti i giochi strizza-cervello che si rispettino (primo tra tutti «Tetris»), il meccanismo di gioco è di una semplicità disarmante: si tratta solamente di aprire delle cassaforti chiuse con un sistema di rotelle contrassegnate da simboli colorati. Per aprire lo sportello, è sufficiente ruotare tutte le manopole in modo che ognuna di esse sia posizionata con i simboli sugli indicatori dello stesso colore. Naturalmente, tutte le rotelle sono a contatto tra loro: il che significa che, come nel cubo di Rubik, il movimento di una di esse influisce su tutte le altre. Non c'è nulla

http://www.oldgamesitalia.net/
http://www.oldgamesitalia.net/

Software Express

http://www.oldgamesitalia.net/

di più frustrante che aver posizionato correttamente tutte le manopole tranne una e scoprire che proprio il movimento di quell'ultima vanifica tutta la fatica fatta in precedenza per spostare correttamente le altre: e questa eventualità si verifica con diabolica frequenza, in «Revelation». Una volta aperta una cassaforte, potete raccogliere i gioielli rinchiusi al suo interno in un piccolo bonus-game, prima di passare al livello seguente.
Graficamente mediocre, «Revelation» entusiasmerà gli appassionati di questo genere di tortura intellettuale e causerà irritazione ed isteria in tutti gli altri. Se non appartenete alla prima categoria di videogiocatori, evitatelo accuratamente.

MIG 29 FULCRUM

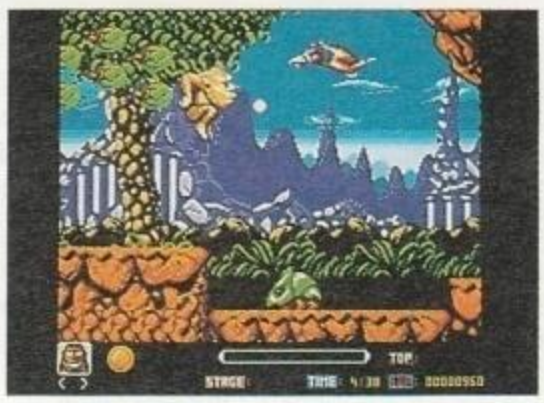
La quantità di simulatori di volo in circolazione è diventata ormai impressionante: «Mig 29» della Domark è l'ultimo, in ordine di apparizione, di questa interminabile serie, e si distingue dai suoi predecessori principalmente per essere il primo simulatore di volo basato su di un

velivolo di produzione sovietica, invece che i soliti F-19 o Tornado.
Essendo il Mig 29 in aereo pressoché sconosciuto nel mondo occidentale, è difficile giudicare il realismo con il quale i suoi comandi e la sua manovrabilità sono stati convertiti su Amiga. Dal punto di vista strettamente ludico, invece, i risultati sono più visibili ed eclatanti: cinque le missioni da portare a termine, l'ultima delle quali non viene rivelata dal manuale essendo accessibile solo dopo aver guadagnato un numero sufficiente di punti nelle precedenti. Il basso numero di missioni è tra le caratteristiche più deludenti di «Mig 29» e non contribuisce certo a prolungarne la

longevità: una volta portate tutte a termine, l'interesse per il gioco diminuisce drasticamente. Per fortuna la grafica e la giocabilità sono molto elevate: in «Mig 29» sono implementate tutte le funzioni ormai obbligatorie in un simulatore di volo degno di questo nome, ed anche qualcuna in più. Nonostante la fluidità ed il realismo dei movimenti, «Mig 29» è più un gioco arcade che un vero simulatore di volo: anche i comandi di volo sono più intuitivi e meno numerosi che nella maggior parte dei programmi di questo tipo.
Sicuramente gli appassionati del genere non mancheranno di aggiungerlo alla propria collezione, ma il titolo di miglior simulatore per Amiga è attualmente ancora vacante. Speriamo che «Falcon II», di imminente uscita, non deluda le aspettative.

TOKI

Una strega malvagia ha rapito la vostra adorata Miho e, per impedirvi di raggiungerla, vi ha trasformato in uno scimmione dall'aspetto ottuso. Non insulteremo la vostra intelligenza rivelandovi



qual è il vostro compito, perché lo avrete sicuramente capito da soli; come pure avrete già intuito che «Toki» è un platform game del tutto tradizionale come giocabilità e meccanismo, nel quale dovete saltare qua e là per tutti i livelli, evitando di fare una brutta fine per mano dei draghi, dei mostri preistorici e delle dozzine di altre creature che infestano la giungla nella quale il gioco è ambientato. Indovinate cosa vi aspetta alla fine di ogni livello? Esatto: l'immancabile super-mostro.
Zero è dunque il voto di «Toki» nella categoria dell'originalità, ma non si può pretendere una trama da film di Spielberg in

una conversione francese (distribuita da Ocean) di un gioco arcade giapponese della misconosciuta Fabtek.
E, a dire il vero, una nota inedita c'è: è la prima volta in cui il cattivo, che rapisce la povera fanciulla amata dal protagonista, è un'altra donna.
Non indagheremo oltre sulle implicazioni di questo fatto, nonostante la nostra fantastia malata tenda a sbizzarrirsi pensando a cosa vorrà mai fare la strega malvagia con la povera Miho...; sottolineiamo invece l'ottimo livello qualitativo della grafica e delle animazioni di «Toki».
Lo scimmione salta su blocchi di ghiaccio, si arrampica su liane, nuota sott'acqua e lancia pure proiettili all'indirizzo dei propri nemici, la cui potenza può essere aumentata raccogliendo i soliti bonus lungo la strada. Un gioco complessivamente molto spettacolare, ma che non sembra destinato ad essere ricordato (e giocato) a lungo.

SUPER MONACO GRAND PRIX

La mancanza di originalità nei giochi di corsa automobilistica per computer è ormai una sgradevole costante: la maggior parte di essi si può liquidare in poche parole, soffermandosi sul livello qualitativo della grafica e sul realismo della simulazione. Pochissimi programmi di questo genere, fino ad ora, sono riusciti ad emergere dalla mediocrità: «Stunt Car Racer» è stato uno di questi, per l'originalità dell'ambientazione ed il puro divertimento dell'azione, come anche «Lotus Esprit», grazie alla possibilità di guidare contemporaneamente in due. «Monaco GP» (U.S. Gold), nonostante la grafica decisamente ottima, non assurge ai livelli di eccellenza dei due programmi sopracitati: si tratta infatti di una simulazione di Formula 1 molto curata ma decisamente poco originale.
Nonostante il titolo, quello di Montecarlo non è fortunatamente l'unico circuito automobilistico sul quale ci si può cimentare: sono selezionabili anche percorsi in Brasile, in Francia ed in Spagna. Inoltre,

ognuno di questi quattro circuiti deve essere completato due volte, in condizioni normali e con la pioggia, per un totale di otto competizioni.
A parte l'ambientazione, il meccanismo di gioco è lo stesso di sempre: occorre completare ogni gara nel minor tempo possibile per evitare di essere squalificati. È invece la grafica a fare la parte del leone in «Monaco GP»; la velocità dell'animazione, la definizione degli sfondi ed anche la presenza dello specchietto retrovisore nella parte superiore dello schermo lo rendono esteticamente superiore alla maggior parte dei giochi di corsa automobilistica in circolazione. Sotto il profilo della giocabilità e della longevità, «Monaco GP» tuttavia non arriva a tagliare per primo il traguardo: è «Lotus Esprit», per ora, a mantenere saldamente il predominio della classifica dei migliori giochi automobilistici.

http://www.oldgamesitalia.net/
http://www.oldgamesitalia.net/

http://www.oldgamesitalia.net/

BRAT

Brat, il protagonista dell'omonimo gioco della Imageworks, è un vero e proprio «enfant terrible»: docile ed angelico di giorno, durante la notte egli sogna di trasformarsi in un piccolo teppista dotato di occhiali da sole e giacca di pelle, e di combinarne di tutti i colori.
Trattandosi di un videogame, queste fantasie

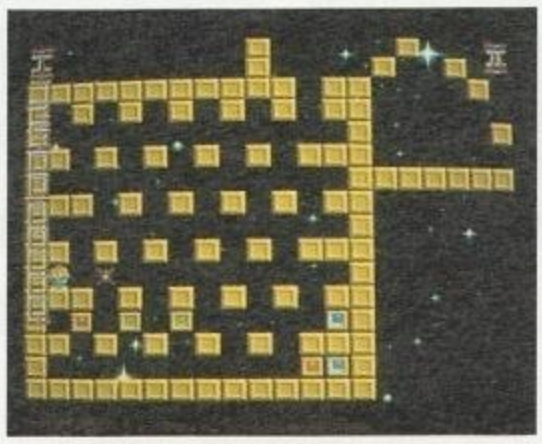
divengono realtà e Brat si ritrova imprigionato in un mondo fantastico, dal quale dovete aiutarlo ad uscire. Oltre 250 schermate compongono i dodici livelli di questo arcade adventure, ambientato rispettivamente in tre tipi diversi di mondo: un parco, lo spazio, ed una città popolata da giocattoli.
Lo schermo mostra l'ambiente che circonda Brat in prospettiva isometrica; con l'aiuto di una serie di icone potete comandare i movimenti della piccola peste e spingerla a

compiere una serie di azioni (raccogliere oggetti, depositarli, costruire ponti, ecc.). La grafica, la musica e lo scrolling sono di livello superlativo.
«Brat» non è però semplice né immediato: oltre al fatto che, come in ogni arcade adventure degno di questo nome, occorrono pazienza e riflessione per completare ogni livello, usando opportunamente gli oggetti e le icone recuperati lungo la strada, «Brat» è reso parecchio complicato dallo strano metodo di controllo dei movimenti del protagonista. Per spostarlo, si fa uso di una serie di icone a forma di freccia: depositando opportunamente queste freccie lungo il percorso, Brat vi si dirigerà raccogliendo gli eventuali oggetti disseminati sulla sua traiettoria.
«Brat» è in definitiva un gioco simpatico ed innovativo, che unisce con successo originalità e giocabilità.

THE POWER

Il nome di questo rompicapo arcade della DemonWare proviene dall'omonimo brano di dance music degli Snap, che funge anche da colonna sonora del gioco. Il protagonista è un personaggio tondo, giallo e sorridente, un parente stretto di PacMan, che potete comandare con il joystick.
Ogni livello è composto da un'area (che può essere ampia anche varie schermate) delimitata da blocchi quadrati e più o meno densamente popolata di ostacoli, ed il vostro compito consiste nel riuscire a raggiungere la vostra amata, raccogliendo tutti i cuoricini sparsi qua e là e saltandole addosso (colta la metafora?).
C'è però un piccolo problema: non potete fermarvi se non andando a sbattere contro un muro o contro un ostacolo. Una volta in

movimento, non potete cambiare direzione né fermarvi a metà strada. La difficoltà consiste quindi nel riuscire a dirigervi verso la meta prestabilita, calcolando opportunamente i movimenti e spostando all'occorrenza i blocchi semoventi presenti in ogni livello.
La meccanica di gioco è praticamente la stessa di «Atomix», un gioco con il quale «The Power» ha abbastanza similitudini da rasentare il plagio.
Oltre che essere atrocemente complicato, «The Power» non è particolarmente attraente. La colonna sonora, completa di voci campionate, è molto curata anche se risulta monotona dopo pochi minuti; la parte grafica, come di consueto in questo genere di giochi, non è invece molto spettacolare.
«The Power» piacerà certamente agli amanti della dance music e dei rompicapo, ma non è abbastanza orginale o coinvolgente per riscuotere successo su vasta scala.



AMIGA PD
MUSIC

SOUND/NOISETRACKER:
I più popolari programmi musicali in TRE DISCHETTI pieni di utility e strumenti campionati.
Lire 20.000

DELTA MUSIC E FUTURE COMPOSER:
Altre due ottime utility sonore, con i relativi demo e strumenti su TRE DISCHETTI.
Lire 20.000

MED 2.12:
Il miglior editor musicale, compatibile con i moduli SoundTracker ma più semplice da usare e interfacciabile MIDI. DIECI DISCHETTI, con utility e centinaia di sample e moduli dimostrativi.
Lire 55.000

Per ricevere i dischetti invia vaglia postale ordinario per l'importo indicato ad AmigaByte, C.so Vitt. Emanuele 15, Milano 20122.

Specifica sul vaglia stesso la tua richiesta ed il tuo indirizzo. Per un recapito più rapido, aggiungi lire 3.000 e richiedi la spedizione espresso!

http://www.oldgamesitalia.net/

http://www.oldgamesitalia.net/

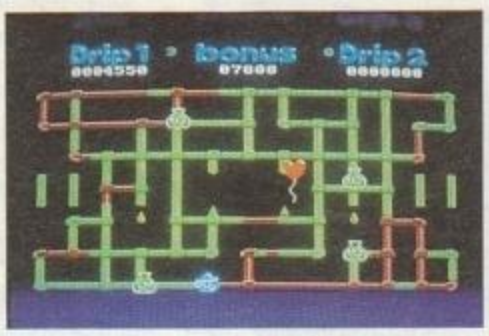
http://www.oldgamesitalia.net/

WANTED

SEI TU
IL PROSSIMO ABBONATO
AD
AMIGA BYTE

Per 11 fascicoli ed altrettanti dischetti direttamente a casa tua

IN PIÙ IN REGALO UN SUPERDISCO
CON DUE SPLENDIDI GIOCHI INEDITI



* *Il superdisco viene inviato anche a chi si abbona a prezzo ridotto (L. 85mila) per ricevere esclusivamente i fascicoli senza dischetto allegato.*

ABBONATI! Cosa aspetti?

Per abbonarti invia vaglia postale ordinario ad Amiga Byte c.so Vitt. Emanuele 15, 20122 Milano.

http://www.oldgamesitalia.net/
http://www.oldgamesitalia.net/